

یک روش استقرار سرویس و تخصیص منابع مبتنی بر الگوریتم جستجوی عروس‌های دریایی ارتقا داده شده برای محیط‌های پردازش لبه همراه چهار لایه‌ای

محمدرضا حقیقت افشار^۱، کامبیز مجیدزاده^{۲*}، محمد مصدري^۳، فرامرز فتح نژاد^۴

^۱ دانشجوی دکترا، گروه کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

^۲ استادیار، گروه کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

^۳ استادیار، گروه کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

^۴ استادیار، گروه کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

مقاله پژوهشی

چکیده

محاسبات لبه همراه (MEC) به عنوان رویکردی جدید در شبکه‌ها، منابع را برای ارائه خدمات باکیفیت در چندین لایه سازمان‌دهی می‌کند. از چالش‌های اساسی در این شبکه‌ها، می‌توان به انتخاب بهینه‌ترین منبع برای سرویس‌های درخواستی کاربران اشاره کرد که با نام استقرار سرویس یا به اختصار SPP شناخته می‌شود. در SPP با در نظر گرفتن محدودیت‌های مختلف QoS، برای هر درخواست کاربران بهینه‌ترین منبع از لایه‌های لبه، مه و ابر انتخاب می‌شود. یک الگوریتم تخصیص بهینه منابع می‌تواند به صورت چشمگیری رضایت کاربران و فراهم‌کنندگان خدمات را افزایش داده و با بهینه کردن ارتباطات و کاهش زمان سرویس‌دهی باعث افزایش بهره‌وری کل سیستم شود. در این مقاله یک چهارچوب جدید با نام EMLJSA برای استقرار سرویس‌ها و تخصیص منابع در شبکه‌های چند لایه‌ای MEC فراهم شده است. در EMLJSA یک نسخه بهبود داده شده از الگوریتم فرا ابتکاری جستجوی عروس‌های دریایی بکار گرفته شده که در آن معادلات پایه این الگوریتم اصلاح شده‌اند. همچنین، یک نسخه از عملگر جهش و یک مکانیزم جستجوی همسایگی آشوبناک برای افزایش بهره‌وری و قابلیت‌های جستجو معرفی گردیده است. برای ارزیابی کارایی الگوریتم پیشنهادی دوازده شبکه MEC چهار لایه‌ای طراحی شده است. نتایج آزمایش‌ها به صورت عددی و گرافیکی با هفت روش مشابه مقایسه شده‌اند، که اثربخشی و برتری روش پیشنهادی را نشان می‌دهند.

تاریخ دریافت:

۱۴۰۳/۸/۱۵

تاریخ پذیرش:

۱۴۰۳/۱۱/۲۷

کلیدواژه‌ها:

محاسبات لبه همراه، پردازش مه و ابر، استقرار سرویس، تخصیص منابع، بهینه‌سازی

نویسنده مسئول:

kambiz.majidzadeh@iau.ac.ir

doi : 10.22034/ABMIR.2025.22356.1071

۱- مقدمه

شبکه‌های اینترنت اشیا (IoT) مبتنی بر ارتباطات اینترنتی به تأخیر حساس هستند و با افزایش حجم داده‌های تولید شده توسط کاربر دچار محدودیت و مشکلاتی می‌شوند [۱]. از جمله این مشکلات می‌توان به محدودیت پهنای باند در ارتباطات راه دور بین دستگاه‌های کاربران و فراهم‌کنندگان سرویس، مصرف انرژی بالا برای ارسال و دریافت داده، وجود تأخیر چشمگیر در برخی کاربردهای بلادرنگ، هزینه‌های بالای تجهیزات و انتقال داده، امکان دسترسی به داده‌ها و نقض حریم خصوصی کاربران و غیره اشاره نمود [۲، ۳]. در نتیجه، محققان الگوی محاسباتی جدیدی بنام محاسبات لبه همراه (Mobile Edge Computing) یا به اختصار MEC توسعه داده‌اند. MEC منابع محاسباتی و ذخیره‌سازی را در نزدیکی کاربران شبکه قرار می‌دهد که باعث کاهش عمده مشکلات ارتباطی و ارائه بهتر خدمات می‌شود [۴، ۵].

در شبکه‌های امروزی پردازش لبه همراه با پردازش مه و ابر ترکیب شده و شبکه‌های جدید فراهم می‌شود که نقاط قوت هر سه نوع پارادایم شبکه را شامل می‌شوند. در معماری جدید چند لایه‌ای، منابع متعددی در نزدیکی کاربران قرار می‌گیرند تا نیازهای بلادرنگ و امنیتی آن‌ها را برطرف سازند. این منابع که منابع لبه نامیده می‌شوند دارای زمان پاسخ کمتر بوده و قدرت‌های پردازشی و ذخیره‌سازی ناچیز دارند [۶]. در کنار این منابع، منابع دیگری در فاصله از دستگاه‌های کاربران قرار می‌گیرند که منابع مه و ابر نامیده می‌شوند. این منابع دارای قدرت پردازش و فضای ذخیره‌سازی بیشتر بوده و برای پردازش‌های سنگین‌تر و دسته‌ای کاربران و همچنین ذخیره‌سازی انبوه آن‌ها مناسب می‌باشند. با این حال، هزینه ارتباطی و استفاده از این منابع و همچنین زمان پاسخ در آن‌ها بیشتر است [۷]. در شبکه‌های ترکیبی حاصله که در ادامه مقاله آن را MEC می‌نامیم کاربران متعددی وجود دارند که سرویس‌های مختلفی را درخواست می‌کنند.

انتخاب بهینه‌ترین منبع از بین منابع متعدد ارائه شده در لایه‌های لبه، مه و ابر برای استقرار سرویس‌های درخواستی کاربران مساله استقرار سرویس (Service Placement Problem) یا به اختصار SPP نامیده می‌شود و یکی از چالش‌های اساسی شبکه‌های MEC

است [۶]. روش‌های متعددی برای حل SPP ارائه شده است که در دسته‌های مختلف سازمان‌دهی می‌شوند. توزیع جغرافیایی دستگاه‌های لبه، تحرک، محدودیت‌های منابع و ماهیت بسیار پویا، مقیاس‌پذیری، تنوع کاربران، خدمات و برنامه‌ها، کیفیت خدمات، کیفیت تجربه، ناهمگونی منابع، استفاده از منابع، محدودیت منابع، استفاده از لینک، محدودیت پهنای باند، امنیت و حریم خصوصی SPP را چالش‌برانگیزتر می‌کند [۸]. SPP از جمله مسائل پیچیده با پیچیدگی محاسباتی غیرخطی است که روش‌های رایج قطعی قادر به حل آن نمی‌باشند به‌خصوص در شبکه‌های مقیاس وسیع با تعداد کاربران زیاد. از این‌رو، در جدیدترین روش‌های ارائه شده برای حل این مساله از الگوریتم‌های بهینه‌سازی فرا ابتکاری استفاده شده است [۹].

الگوریتم‌های فرا ابتکاری جدیدترین رویکرد یادگیری افزایشی برای حل مسائل پیچیده دنیای واقعی می‌باشند که می‌توانند مسئله مورد نظر را با دقت قابل قبول و در زمان معقول حل نمایند [۱۰]. این الگوریتم‌ها برای حل مساله مورد نظر نیاز به هیچ اطلاعاتی نداشته و صرفاً براساس آزمون و خطا و کمینه/بیشینه کردن مقدار تابع هدف پاسخ بهینه را تولید می‌کنند [۱۱]. در تابع هدف این الگوریتم‌ها می‌توان بازه وسیعی از معیارهای تعریف‌شده در حوزه مورد نظر را بکار برد [۱۲].

الگوریتم‌های فرا ابتکاری دارای دو قابلیت عمده برای جستجوی فضای مساله می‌باشند. اولی قابلیت اکتشاف (exploration) است که به‌واسطه آن الگوریتم تلاش می‌کند حجم بیشتری از فضای مساله را کاوش نموده و مناطق امیدوارکننده را پیدا کند [۱۳]. دومین قابلیت، بهره‌برداری (exploitation) است که با کمک آن الگوریتم مناطق امیدوارکننده پیدا شده تاکنون را برای پیدا کردن راه‌حل‌های بهتر جستجو می‌نماید [۱۴-۱۶].

ارائه روش‌های مبتنی بر الگوریتم‌های فرا ابتکاری برای حل مساله SPP و بررسی تأثیر معیارهای مختلف بر روش کارایی شبکه‌های MEC یکی از موضوعات محبوب پژوهشی در این حوزه بوده که به‌طور فزاینده‌ای در حال افزایش است. لذا در این پژوهش محققان با بهره‌گیری از نقاط قوت الگوریتم‌های فرا ابتکاری یک روش

پاسخ، توان عملیاتی و هزینه محاسباتی بوده که برتری روش پیشنهادی بر روش‌های مشابه را نشان دادند.

به‌طور مشابه Nethaji و Chidambaram یک روش تخصیص منابع جدید به نام DGW-SBDR برای محیط‌های محاسباتی مه ارائه کردند که قابلیت تعادل بار در محیط‌های محاسباتی را دارا است [۱۸]. در الگوریتم DGW-SBDR، الگوریتم تکامل تفاضلی، بهینه‌ساز گرگ‌خاکستری، گرایانت تصادفی بلمن و یادگیری تقویتی عمیق مورد استفاده قرار گرفته‌اند. در روش پیشنهادی بهینه‌ساز گرگ‌خاکستری توسط معادلات الگوریتم تکامل تفاضلی تقویت شده و به منظور استقرار وظایف کاربران بر روی ماشین‌های مجازی فراهم شده استفاده شده است. در انتخاب منبع برای وظایف کاربران الزامات و محدودیت‌های تعریف شده توسط کاربران مورد توجه قرار گرفته است. همچنین، الگوریتم ترکیبی گرگ‌خاکستری و تکامل تفاضلی توانایی تشخیص استفاده استقرار بیش از حد وظایف به یک منبع یا تخصیص وظایف کم به منابع را داشته و در صورت رخداد این مسائل، وظایف را از منابع شلوغ به منابع خلوت منتقل می‌کند. درحالی که ماشینی هیچ وظیفه‌ای برابر پردازش نداشته باشد ممکن است برای کاهش تأخیر خاموش شوند. در گام بعدی، الگوریتم گرایانت تصادفی بلمن و یادگیری افزایشی عمیق منابع بهینه را انتخاب و به وظایف تخصیص می‌دهند. روش پیشنهادی در محیط iFogSim با زبان جاوا پیاده‌سازی شده و نتایج حاصله با الگوریتم تخصیص منابع پویا انرژی کارآمد و الگوریتم تخصیص منبع متعادل بار برای دروازه‌های مه مقایسه شده است.

علاوه بر این، کومار و همکاران الگوریتم جستجوی کلاغ را بکار گرفته و یک روش جدید برای تخصیص منابع و پیش‌بینی بار کاری در شبکه‌های اینترنت اشیا صنعتی مبتنی بر مه ارائه کرده‌اند [۱۹]. روش پیشنهادی شامل یک فاز تجزیه و تحلیل است که بار کاری را با استفاده از الگوریتم رمزگشای خودکار عمیق پیش‌بینی می‌کند. سپس این فاز گره‌های مه را با توجه به تقاضای بار کاری شبکه اینترنت اشیا صنعتی مقیاس‌دهی می‌کند. در مرحله بعدی، الگوریتم جستجوی کلاغ بهینه‌ترین گره‌های مه را برای جریان‌های کاری انتخاب می‌کند. در روش پیشنهادی در این پژوهش، معیارهای

جدید با نام EMLJSA برای حل مساله SPP در محیط‌های پردازش لبه همراه ترکیب شده با محاسبات مه و ابر ارائه داده‌اند. در روش پیشنهادی، معادلات حرکتی الگوریتم عروس‌های دریایی که یک الگوریتم بهینه‌سازی قدرتمند است، اصلاح و در راستای مساله SPP بهبود داده شده‌اند. همچنین، برخی از پارامترها و محاسبات این الگوریتم با هدف کاهش بار محاسباتی حذف شده است. در ادامه، یک نسخه از عملگر جهش با الگوریتم عروس‌های دریایی اصلاح شده ترکیب شده است. همچنین، مکانیزمی برای بهبود هرچه بیشتر بهترین راه‌حل یافته شده در الگوریتم EMLJSA فراهم شده است. در نهایت، دوازده شبکه MEC چهار لایه‌ای (اینترنت اشیا، لبه، مه و ابر) با مشخصه‌های متفاوت طراحی و برای ارزیابی عملکرد الگوریتم‌ها استفاده شده است.

۲- پیشینه پژوهش

در زمینه استقرار سرویس یا تخصیص منابع در محیط‌های پردازش لبه، مه و ابر رویکردهای متفاوتی اتخاذ شده و روش‌های مختلفی پیشنهاد شده است. این بخش از مقاله مرور مختصری از روش‌های ارائه شده مبتنی بر الگوریتم‌های فرا ابتکاری را فراهم می‌کند. در پژوهشی که توسط سلیمیان و همکاران انجام شده است، یک روش جدید تخصیص منابع برای محیط‌های مه - لبه ارائه شده است [۱۷]. در روش پیشنهادی از بهینه‌ساز گرگ‌خاکستری برای قراردادن سرویس خودکار در محیط‌های محاسباتی استفاده شده است. هدف اصلی نویسندگان در این مقاله، تخصیص وظایف اینترنت اشیا به بهینه‌ترین گره‌های مه بوده است که منجر به بهبود عملکرد کلی سیستم و کاهش زمان اجرا می‌شود. در روش پیشنهادی گره‌های مه مستقر شده به صورت ناهمگن تعریف شده و برای هر کدام قابلیت‌ها و محدودیت‌های متفاوت در نظر گرفته شده است. روش تخصیص منابع ارائه شده در این پژوهش دارای سه لایه بوده و نحوه تعامل گره‌های مه با دستگاه‌های اینترنت اشیا را به‌طور آشکار مدل‌سازی می‌کند. روش پیشنهادی توسط زبان متلب شبیه‌سازی شده و نتایج به دست آمده توسط آن با نتایج الگوریتم بهینه‌سازی ملخ، بهینه‌سازی ازدحام ذرات، الگوریتم خفاش و الگوریتم ژنتیک مقایسه شده است. مقایسه‌های انجام شده بین روش پیشنهادی و الگوریتم‌های رقیب بر اساس معیارهای زمان

عمده‌ترین کاستی روش‌های پیشین این است که ترکیب مناسبی از لایه‌های مختلف را ارائه نکرده‌اند یا تنها دو لایه از سه لایه را بکار برده‌اند. در کنار این موارد، جنبه بسیار مهم مقیاس‌پذیری به‌صورت بسیار ضعیف در پژوهش پیشین مطالعه شده است.

لذا در این پژوهش سعی شده است مکانیزم‌های جدیدی در توسعه یک روش استقرار منابع معرفی گردد. همچنین، از معیارهای ارزیابی مناسبی استفاده شده است تا کارایی الگوریتم پیشنهادی به‌طور مناسبی ارزیابی شود. در کنار این‌ها، نتایج روش پیشنهادی با نتایج الگوریتم‌های متعددی مقایسه شده است تا برتری آن به صورت دقیق‌تری آشکار شود. علاوه بر این، در چهارچوب پیشنهادی از هر سه لایه لایه، مه و ابر استفاده شده و سیاست‌های به‌کارگیری منابع هر کدام از لایه‌ها مشخص شده است. همچنین، مکانیزمی پویا و مقیاس‌پذیر برای تخصیص منبع، استقرار سرویس و زمان‌بندی وظایف در چهارچوب پیشنهادی فراهم شده است.

۳- الگوریتم جستجوی عروس دریایی

الگوریتم بهینه‌سازی جستجوی عروس دریایی (Jellyfish Search Algorithm) یا به اختصار JSA یکی از جدیدترین و قوی‌ترین الگوریتم‌های فرا ابتکاری الهام گرفته شده از طبیعت است [۲۱]. اساس کار این الگوریتم براساس زندگی عروس‌های دریایی در طبیعت است که به صورت ریاضی مدل‌سازی شده و برای کاربردهای بهینه‌سازی ارائه شده است. در این الگوریتم چهار مفهوم جدید معرفی شده است که عبارت‌اند از: جریان اقیانوس، حرکت غیرفعال، حرکت فعال و مکانیزم کنترل زمان که در ادامه معرفی شده و نحوه فرموله شدن آن‌ها در الگوریتم JSA بیان می‌گردد.

در الگوریتم JSA دسته عروس‌های دریایی به صورت تصادفی با استفاده از تئوری آشوب تولید می‌شود. معادله (۱) نحوه تولید یک دسته از عروس‌های دریایی را به صورت ریاضی بیان می‌کند.

$$X_i = LB + CV \times (UB - LB) \quad (1)$$

در معادله (۱)، X_i موقعیت عروس دریایی i ام، lb و ub به ترتیب کران‌های پایین و بالای فضای مسئله است و CV مقدار آشوبناکی است که توسط تابع آشوبناک Logistic به‌دست آمده است. ضرب شدن مقدار تابع آشوبناک که مقداری بین صفر و یک دارند، در

هزینه اجرا و تأخیر به‌عنوان توابع هدف در نظر گرفته شده‌اند. الگوریتم پیشنهادی در شبیه‌ساز iFogSim پیاده‌سازی شده و نتایج به‌دست آمده با بهینه‌ساز گرگ خاکستری، بهینه‌سازی ازدحام ذرات، الگوریتم بهینه‌سازی خفاش و کلنی زنبور مصنوعی از نظر هزینه اجرا، زمان پاسخ، توان عملیاتی و نسبت رد درخواست مقایسه شده‌اند. نتایج آزمایش‌ها برتری روش پیشنهادی بر الگوریتم‌های رقیب را نشان دادند.

در پژوهشی دیگر، اصغری و همکاران از الگوریتم بهینه‌سازی نهنگ و تئوری بازی استفاده کرده و یک الگوریتم چند هدفه جدید بنام MES-WG برای تخصیص سرویس‌دهنده‌های لایه به دستگاه‌های تلفن همراه ارائه دادند [۲۰]. در مرحله نخست الگوریتم MES-WG، منطقه‌ای که دستگاه‌های سیار در آن توزیع شده‌اند به مناطق فرعی تقسیم می‌شود که هدف از این کار کاهش پیچیدگی محاسباتی بوده است. در مرحله دوم، مکان بهینه سرویس‌دهنده‌های لایه توسط الگوریتم بهینه‌سازی نهنگ تعیین می‌شود. در فاز بعدی الگوریتم، سرویس‌دهنده‌ها توسط یک شبکه عصبی به صورت محلی در زیر مناطق تقسیم‌شده مستقر می‌شوند. در فاز چهارم روش پیشنهادی، نرخ همگرایی تخصیص منابع به وظایف در تمام زیر مناطق توسط تئوری بازی افزایش می‌یابد. روش پیشنهادی در محیط متلب پیاده‌سازی شده و نتایج آن با الگوریتم ژنتیک، بهینه‌ساز گرگ خاکستری، الگوریتم تخصیص سرویس‌دهنده اکتشافی مبتنی بر K میانگین و الگوریتم یادگیری افزایشی چندعاملی مقایسه شده است.

همان‌طور که مشاهده می‌شود الگوریتم‌های بهینه‌سازی بکار گرفته شده در روش‌های موجود که برخی از برجسته‌ترین آن‌ها در این بخش کاوش شدند الگوریتم‌های قدیمی است. در حالیکه، محیط‌های امروزی نیازمند به‌کارگیری بررسی تأثیر الگوریتم‌های جدید است. علاوه بر این، معیارهای ارزیابی استفاده شده در روش‌های پیشین ناکافی بوده و برای نشان دادن کارآمدی یک سیستم تخصیص منابع پیچیده مناسب نیست. به‌طور مشابه مشاهده می‌گردد که تعداد الگوریتم‌های رقیب در پژوهش‌های صورت گرفته کم است. در صورتیکه برای نشان دادن برتری یک الگوریتم جدید، مقایسه‌ها باید با الگوریتم‌های بیشتری انجام شود. با این حال،

در معادله $X_i, (\xi)$ موقعیت عروس دریایی i ام در جمعیت فعلی، r_2 عددی است که به‌طور تصادفی از بازه $(0,1)$ انتخاب شده است. این معادله باعث تغییر موقعیت فعلی عروس دریایی در راستای جهت تعیین‌شده با Dir (جهت جریان اقیانوسی) می‌شود که به صورت زیر محاسبه می‌شود:

$$Dir = \frac{1}{N} \sum (X_B - e_c \cdot X_i) = X_B - e_c \times \frac{\sum X_i}{N} \quad (5)$$

$$= X_B - e_c \cdot \mu \quad (6)$$

در معادلات (۵) و (۶)، N تعداد عروس‌های دریایی موجود در جمعیت، X_B موقعیت بهترین عروس دریایی جمعیت، e_c عامل حاکم بر جاذبه، $\beta = 3$ ضریب توزیع و μ پارامتری است که توسط معادله (۷) محاسبه می‌شود. جهت جریان اقیانوسی (Dir) معیاری است که در الگوریتم JSA معرفی گردیده تا جهت جابجایی بردار موقعیت عروس‌های دریایی تعیین گردد.

$$\mu = \frac{1}{N} \sum_{i=1}^N X_i \quad (7)$$

در حالت کلی معادله (۵) می‌تواند به صورت زیر بازنویسی شود:

$$Dir = X_B - df \quad (8)$$

$$df = e_c \cdot \mu \quad (9)$$

بر اساس فرض توزیع فضایی نرمال می‌توان چنین اظهار کرد که عروس‌های دریایی در اطراف میانگین موقعیت تمام عروس‌های دریایی موجود در جمعیت پراکنده می‌شوند. در نتیجه، می‌توان df را به صورت زیر تعریف نمود:

$$df = \beta \cdot \sigma \cdot r_4 \quad (10)$$

در معادله (10) ، r_4 یک عدد تصادفی در بازه $(0,1)$ بوده و σ به صورت زیر به دست می‌آید:

$$\sigma = r_4 \cdot \mu \quad (11)$$

با توجه به معادلات (۶) و (۹)، جهت جریان اقیانوس (Dir) به صورت زیر در JSA تعریف گردیده است:

$$Dir = X_B - r_3 \cdot \beta \times \mu \quad (12)$$

در ادامه، زمانی که مقدار پارامتر کنترل زمان کوچک‌تر از حد آستانه باشد ($C < C_0$)، عروس دریایی دو نوع حرکت از خود نشان می‌دهد که حرکات غیرفعال و فعال می‌باشند. توسط حرکت غیرفعال که

بازه فضای مساله باعث می‌گردد که هر بعد موقعیت عروس‌های دریایی عددی بین کران پایین و کران بالای فضای مساله باشد. معادله (۲) نگاشت آشوبناک لجستیک را ارائه می‌دهد.

$$CV(t+1) = \alpha \cdot CV(t) \times (1 - CV(t)) \quad (2)$$

قابل ذکر است که تابع آشوبناک Logistic باید یک مقدار اولیه $(CV(0))$ داشته باشد که بتواند دنباله بعدی را ایجاد کند. مقدار اولیه این تابع باید عددی بین صفر و یک باشد ولی نمی‌تواند برابر با ۰، ۰/۲۵، ۰/۵، ۰/۷۵ و ۱ باشد زیرا در این صورت مقدار تابع آشوبناک در طی زمان ثابت خواهد بود. همچنین α یک ضریب است که مقدار آن برابر چهار است.

بعد از اینکه دسته اولیه عروس‌های دریایی ایجاد و موقعیت اولیه عروس‌های دریایی توسط معادله (۲) تعیین گردید، پارامتری به نام کنترل زمان (C) محاسبه می‌شود تا نوع حرکت و جابجایی عروس دریایی مشخص شود. به عبارتی دیگر، عروس‌های دریایی توسط پارامتر کنترل زمان تصمیم می‌گیرند که جریان اقیانوس را دنبال کرده یا حرکتی در داخل جمعیت داشته باشند. از آنجایی که جریان‌های اقیانوسی حاوی مقدار قابل‌توجهی از منابع غذایی هستند، عروس‌های دریایی به آن سمت حرکت می‌کنند. باین حال، مقدار منابع غذایی در جریان‌های اقیانوسی به‌مرور زمان توسط عروس‌های دریایی مصرف‌شده و به اتمام می‌رسد در نتیجه منجر به مهاجرت عروس‌های دریایی به جریان‌های اقیانوسی دیگر می‌شود. این پدیده توسط پارامتر کنترل زمان در JSA شبیه‌سازی شده و محاسبه آن به صورت ریاضی در معادله (۳) فراهم شده است.

$$C = \left| \left(1 - \frac{t}{T} \right) \times 2 \cdot r_1 - 1 \right| \quad (3)$$

در معادله (۳)، t تکرار فعلی الگوریتم، T حداکثر تعداد تکرار الگوریتم و r_1 یک عدد تصادفی در بازه $[0,1]$ است. همراه با پارامتر کنترل زمان، پارامتر تصمیم دیگری به نام C_0 در JSA تعریف شده است که نقش بسزایی در تعیین نوع حرکت عروس‌های دریایی دارد. مقدار پیش‌فرض و پیشنهادشده برای C_0 مقدار ۰/۵ است. در الگوریتم JSA زمانی که $C \geq C_0$ باشد، عروس دریایی جریان اقیانوس را دنبال کرده و موقعیت خود را با استفاده از معادله (۴) به‌روزرسانی می‌کند.

$$X_i = X_i + r_2 \cdot Dir \quad (4)$$

در غیر این صورت:

اگر $rand() \geq 1 - C$:

عروس دریایی حرکت غیرفعال انجام

می‌دهد

X_i را توسط معادله (۱۳) به‌روزرسانی کن

در غیر این صورت:

عروس دریایی حرکت فعال انجام می‌دهد

X_i را توسط معادله (۱۴) به‌روزرسانی کن

کران‌های X_i را بررسی کن

F_i را محاسبه کن

F_i و X_i را به‌روزرسانی کن

F_i و X_i را برگردان

در این معادلات، X_j موقعیت مکانی عروس دریایی است که به صورت تصادفی به‌عنوان مقصد انتخاب شده و F_j میزان منابع غذایی موجود در مقصد است. شبه کد الگوریتم JSA در شکل (۲) بیان گردیده است.

۴- مدل شبکه و فرموله کردن مساله

این بخش از مقاله مدل شبکه اتخاذ شده در پژوهش فعلی را معرفی کرده و مساله استقرار سرویس‌ها در محیط‌های پردازش لبه ترکیب شده با پردازش مه و ابر را تعریف می‌کند. همچنین، این بخش نحوه فرموله کردن مساله استقرار سرویس‌ها به‌عنوان یک مساله بهینه‌سازی را فراهم می‌کند. علاوه بر این، فرمول ریاضی تابع هدف استفاده شده در روش پیشنهادی در این بخش ارائه می‌گردد.

در شبکه MEC مورد استفاده در این مقاله، فرض شده است که N_I دستگاه اینترنت اشیا در ناحیه شبکه مستقر شده‌اند [۶، ۲۲، ۲۳]. این دستگاه‌ها به صورت $I = \{i_1, i_2, \dots, i_{N_I}\}$ نشان داده شده‌اند. از آنجایی که تحرک در لایه دستگاه‌های اینترنت اشیا مجاز است، مختصات هر دستگاه اینترنت اشیا (i_i) می‌تواند به‌طور متناوب تغییر کند. بنابراین، مختصات دستگاه‌های اینترنت اشیا را می‌توان در هر شکاف زمانی به صورت زیر تعیین کرد:

$$PI = \begin{bmatrix} X_{i_1} & Y_{i_1} \\ X_{i_2} & Y_{i_2} \\ \vdots & \vdots \\ X_{i_{N_I}} & Y_{i_{N_I}} \end{bmatrix} \quad (16)$$

به صورت ریاضی در معادله (۱۳) بیان شده است عروس دریایی در اطراف موقعیت فعلی خود حرکت می‌کند.

$$X_i = X_i + \gamma \cdot r_5 \times (ub - lb) \quad (13)$$

در معادله (۱۳)، γ پارامتری است که طول حرکت را مشخص می‌کند. مقدار پیشنهاد شده برای γ در مقاله پایه 0.1 است.

مطالعات صورت گرفته بر روی رفتار عروس‌های دریایی در طبیعت نشان می‌دهند که با گذشت زمان، عروس‌های دریایی به‌جای حرکت غیرفعال، حرکت فعال رو انتخاب و انجام می‌دهند. در حرکت فعال، عروس دریایی موقعیت خود را نسبت به موقعیت عروس دریایی دیگر در جمعیت به‌روزرسانی می‌کند. در این حالت، هر عروس دریایی یک عروس دریایی دیگر را به صورت تصادفی از جمعیت فعلی به‌عنوان مقصد خود انتخاب می‌کند. سپس، عروس دریایی منابع غذایی موجود در اطراف مقصد را ارزیابی می‌کند. اگر منابع غذایی در نقطه مقصد بیشتر از موقعیت فعلی باشد، عروس دریایی به سمت مقصد حرکت می‌کند. در غیر این صورت، عروس دریایی برخلاف موقعیت مقصد حرکت می‌کند. این نوع حرکت عروس‌های دریایی (حرکت فعال) توسط معادله (۱۴) به صورت ریاضی مدل‌سازی شده است.

$$X_i = X_i + r_5 \cdot step \quad (14)$$

$$step = \begin{cases} X_i - X_j F_j \geq F_i \\ X_j - X_i F_j < F_i \end{cases} \quad (15)$$

الگوریتم (۱): شبه کد الگوریتم JSA

ورودی‌ها: ub, lb, dim, CV_0, T, N

خروجی‌ها: F_B و X_B

برای هر عروس دریایی در بازه $i \in [1, N]$:

عروس دریایی i ام (X_i) را با معادله (۱) مقداردهی اولیه کن

برازندگی عروس دریایی i ام (F_i) را محاسبه کن

بهترین عروس دریایی (X_B) را از جمعیت اولیه پیدا کن

تکرار فعلی (t) را برابر یک کن

تا زمانی که $t \leq T$ است:

برای هر عروس دریایی در بازه $i \in [1, N]$:

پارامتر C را توسط معادله (۳) محاسبه کن

اگر $C \geq C_0$:

عروس دریایی جریان اقیانوس را دنبال می‌کند

X_i را توسط معادله (۴) به‌روزرسانی کن

هر دستگاه اینترنت اشیا در لایه اینترنت اشیا می‌تواند یک یا چند سرویس ارائه شده در شبکه را درخواست کند. در این مقاله، مجموعه خدمات درخواست شده توسط دستگاه‌های اینترنت اشیا به صورت $S = \{S_1, S_2, \dots, S_{N_S}\}$ نشان داده شده است. هر سرویس درخواست شده $S_i \in S$ به سطوح مختلفی از کیفیت خدمات نیاز دارد که به صورت زیر بیان می‌شود:

در معادله (۱۷)، G_i ، i امین دستگاه اینترنت اشیا است که سرویس را درخواست کرده است، R_i^{CPU} ، R_i^{RAM} و R_i^{DISK} به ترتیب مقادیر CPU، RAM و دیسک مورد نیاز برای پردازش سرویس مورد نظر هستند. همچنین S_i اندازه سرویس در مقیاس میلیون دستورالعمل (MI) است و DL_i مهلت زمانی یا حداکثر زمانی است که پردازش سرویس i ام باید خاتمه یابد.

در سیستم بکارگرفته شده در پژوهش پیش‌رو، برخی از منابع فراهم شده در شبکه در نزدیکی دستگاه‌های اینترنت اشیا و منشأ درخواست سرویس‌ها به منظور ارائه خدمات ضروری قرار گرفته‌اند که سرورهای لبه نامیده شده‌اند. این سرورهای لبه به اندازه کافی قدرتمند نیستند و به صورت $E = \{e_1, e_2, \dots, e_{N_{ES}}\}$ نشان داده می‌شوند که N_{ES} تعداد سرورهای لبه است. مشخصات سرور لبه i ام (e_i) به صورت زیر بیان می‌شود:

در معادله (۱۸)، که در آن e_i^{CPU} قدرت محاسباتی سرور لبه i ام با مقیاس میلیون دستورالعمل در ثانیه است (MIPS)، e_i^{BW} پهنای باند اختصاص داده شده به e_i ، e_i^{ECR} نرخ مصرف انرژی (برای یک ثانیه پردازش) است. e_i^{RAM} و e_i^{DISK} به ترتیب ظرفیت RAM و دیسک e_i هستند. همچنین موقعیت سرورهای لبه با یک ماتریس به صورت زیر نمایش داده می‌شود:

علاوه بر این، N_{FN} تا گره‌های میانه در لایه میانه سیستم بکار گرفته شده مستقر شده‌اند که به صورت $F = \{f_1, f_2, \dots, f_{N_{FN}}\}$ نشان داده می‌شوند. در کنار این، N_{PM} ماشین‌های پردازشی در لایه ابر نیز در سیستم مورد نظر تعبیه شده‌اند که به صورت

$$PE = \begin{bmatrix} X_{e_1} & Y_{e_1} \\ X_{e_2} & Y_{e_2} \\ \vdots & \vdots \\ X_{e_{N_{ED}}} & Y_{e_{N_{ED}}} \end{bmatrix} \quad (۱۹)$$

یک روش استقرار سرویس و تخصیص منابع مبتنی بر الگوریتم جستجوی عروس‌های دریایی ارتقا داده شده برای محیط‌های پردازش لبه همراه چهار لایه‌ای

بیش از سرورهای لبه و کمتر از ماشین‌های پردازشی است. ماشین‌های پردازشی فراهم شده در لایه ابری قوی‌ترین منابع هستند. مشخصات گره‌های میانه و ماشین‌های پردازشی به ترتیب با معادلات (۲۰) و (۲۱) نشان داده شده‌اند.

$$f_i = \{f_i^{CPU}, f_i^{RAM}, f_i^{DISK}, f_i^{BW}, f_i^{ECR}\} \quad (۲۰)$$

$$c_i = \{c_i^{CPU}, c_i^{RAM}, c_i^{DISK}, c_i^{BW}, c_i^{ECR}\} \quad (۲۱)$$

در الگوریتم پیشنهادی، موقعیت گره‌های میانه و ماشین‌های پردازشی ابری ثابت بوده و به ترتیب با معادلات (۲۲) و (۲۳) نشان داده شده است.

$$PF = \begin{bmatrix} X_{f_1} & Y_{f_1} \\ X_{f_2} & Y_{f_2} \\ \vdots & \vdots \\ X_{f_{N_{FN}}} & Y_{f_{N_{FN}}} \end{bmatrix} \quad (۲۲)$$

$$PC = \begin{bmatrix} X_{c_1} & Y_{c_1} \\ X_{c_2} & Y_{c_2} \\ \vdots & \vdots \\ X_{c_{N_{PM}}} & Y_{c_{N_{PM}}} \end{bmatrix} \quad (۲۳)$$

با توجه به این نکته که منابع مورد نیاز سرویس‌ها و قابلیت‌های ذخیره‌سازی و محاسباتی سرورهای لبه، میانه و ابر یا به‌طور کلی منابع مشخص، ثابت و محدود هستند، سرویس‌های درخواست شده توسط کاربران فقط می‌توانند بر روی زیرمجموعه‌ای از منابع قرار گیرند. در نتیجه، متغیر تصمیم X_i^j برای تخصیص منبع یا استقرار سرویس را می‌توان به صورت زیر تعریف کرد:

$$X^{S_i} \in \{1, 2, \dots, N_l, N_l + 1, N_l + 2, \dots, (N_l + N_{FN}), (N_l + N_{FN}) + 1, \dots, (N_l + N_{FN} + N_{PM})\}, \quad (۲۴)$$

$$i = 1, 2, \dots, N_S$$

همان‌طور که در معادله (۲۴) نیز نشان داده شده است، منابع لبه، میانه و ابر در دسترس در مخزنی به نام R جمع‌آوری شده‌اند. سپس هر S_i به یک منبع مناسب (r_j) در R اختصاص داده می‌شود. بنابراین، معادله (۲۴) را می‌توان به صورت زیر بازنویسی کرد:

$$X^i = r_j, \text{ where } i = 1, 2, \dots, N_S \text{ and } r_j \in R \quad (۲۵)$$

با استفاده از فرموله کردن پیشنهادی، سرویس درخواست شده i ام (S_i) را می‌توان به یکی از منابع موجود در لایه لبه، میانه یا ابر اختصاص داد. براین اساس، یک راه‌حل (X_i) را می‌توان برای ایجاد یک طرح استقرار سرویس‌ها و تخصیص منابع به آن‌ها

ام و RT_{S_i} زمان پاسخ S_i است. همچنین R_k^{CPU} ، R_k^{RAM} و R_k^{DISK} به ترتیب CPU، RAM، دیسک مورد نیاز سرویس i ام می‌باشند. هنگامی که طرح نگاشت سرویس‌ها به منابع مشخص گردید و محدودیت‌های تعریف‌شده برآورده شد، مقدار تابع هدف از طریق رابطه (۳۰) محاسبه می‌شود.

استفاده کرد. یک راه‌حل نمونه در روش فرموله کردن پیشنهادی در شکل (۱) نشان داده شده است. با این وجود، هر راه‌حل و نقشه‌نگاشتی که توسط معادله (۲۵) به دست آید قابل قبول نیست و باید محدودیت‌های CPU، RAM، دیسک و مهلت بیان‌شده در معادلات (۲۶) تا (۲۹) را برآورده کند. در این معادلات، R مخزن منابع قابل استفاده در لایه‌های لبه، مه و ابر، X_i^k بعد k ام راه‌حل i

S_9	...	S_{39}	S_1	S_5	S_{20}	S_{36}	S_{11}	S_8	S_{10}	S_{23}	S_{42}	سرویس
r_{18}	...	r_1	r_{16}	r_{16}	r_{10}	r_5	r_5	r_6	r_7	r_{21}	r_{10}	منبع

شکل (۱): یک راه‌حل نمونه در روش پیشنهادی برای استقرار سرویس‌ها

شروع پردازش سرویس S_i بر روی منبع r_j است. زمانی که هیچ سرویسی در حال پردازش بر روی منبع r_j نباشد، زمان شروع صفر خواهد بود. با این حال، اگر منبع r_j در حال پردازش سرویس(های) دیگری باشد، زمان شروع سرویس S_i زمان تکمیل سرویس(های) قبلی اختصاص داده شده به منبع r_j خواهد بود. همچنین، PT_{S_i} زمان پردازش S_i است که به اندازه S_i و قابلیت پردازشی منبع اختصاص داده شده بستگی دارد. E_p مقدار انرژی مصرف‌شده برای پردازش همه سرویس‌ها، BT_{r_i} زمانی است که منبع r_j در حال پردازش بوده است و r_i^{ECR} نرخ مصرف انرژی منبع r_j است. r_i^{BW} پهنای باند اختصاص داده شده به r_i ، T_p قدرت انتقال، d_{S_i, r_j} فاصله بین دستگاه درخواست کننده سرویس S_i و منبع اختصاص داده شده به آن r_j ، و θ و ρ ضرایبی به ترتیب با مقادیر ۱ و ۱۰۰ هستند. علاوه بر این، α ، β و γ پارامترهایی هستند که اهمیت زمان تکمیل پردازش سرویس‌ها، انرژی پردازشی و انرژی ارتباطی را مشخص می‌کنند و به ترتیب برابر $0/4$ ، $0/3$ و $0/3$ تنظیم شده‌اند. از آنجایی که واحد مقیاس‌های T و E متفاوت هستند، از معادلات (۴۰) تا (۴۲) برای حذف مقیاس یا نرمال‌سازی T ، E_p و E_C قبل از جمع بستن آن‌ها استفاده شده است.

$$NT = \frac{T - T_{min}}{T_{max} - T_{min}} \quad (40)$$

$$NE_p = \frac{E_p - E_{pmin}}{E_{pmax} - E_{pmin}} \quad (41)$$

$$NE_C = \frac{E_C - E_{Cmin}}{E_{Cmax} - E_{Cmin}} \quad (42)$$

$$\sum_{r_j \in R} \sum_{k=1}^{N_s} (RP_i^k == r_j) \times R_k^{CPU} \leq r_j^{CPU} \quad (26)$$

$$\sum_{r_j \in R} \sum_{k=1}^{N_s} (RP_i^k == r_j) \times R_k^{RAM} \leq r_j^{RAM} \quad (27)$$

$$\sum_{r_j \in R} \sum_{k=1}^{N_s} (RP_i^k == r_j) \times R_k^{DISK} \leq r_j^{DISK} \quad (28)$$

$$RT_{S_i} \leq DL_i; \forall S_i \in S \quad (29)$$

$$F_i = \alpha \cdot T + \beta \cdot E_p + \gamma \cdot E_C \quad (30)$$

$$T = \sum_{S_i \in S} RT_{S_i} \quad (31)$$

$$RT_{S_i} = CT_{S_i} + ST_{S_i} + PT_{S_i} \quad (32)$$

$$CT_{S_i} = \frac{S_i}{r_j^{BW}} \quad (33)$$

$$PT_{S_i} = \frac{S_i}{r_j^{CPU}} \quad (34)$$

$$E_p = \sum_{r_i \in R} BT_{r_i} \times r_i^{ECR} \quad (35)$$

$$E_C = \sum_{S_i \in S} T_p \times \frac{S_i}{R} \quad (36)$$

$$R = \theta \times r_i^{BW} \times \log_2 \left(1 + \frac{T_p \times G}{\rho} \right) \quad (37)$$

$$G = 127 + 30 \times \log(d_{S_i, r_j}) \quad (38)$$

$$d_{S_i, r_j} = \sqrt{(X_{S_i} - X_{r_j})^2 + (Y_{S_i} - Y_{r_j})^2} \quad (39)$$

در معادلات (۳۰) تا (۳۹)، F_i مقدار تابع هدف راه‌حل i ام، T زمانی است که پردازش همه خدمات تکمیل می‌شود، RT_{S_i} زمانی است که پردازش S_i به پایان می‌رسد، CT زمان ارتباطات، و ST_{S_i} زمان

۵- روش پیشنهادی

این مقاله یک روش جدید بنام Enhanced Mutation and Levy flight-based Jellyfish Search Algorithm (EMLJA) برای استقرار سرویس‌ها به صورت بهینه در منابع فراهم شده در لایه‌های لبه، مه و ابر پیشنهاد می‌کند. روش پیشنهادی براساس الگوریتم بهینه‌سازی فرا ابتکاری جستجوی عروس‌های دریایی (JSA) است. با این حال، تغییراتی بر روی الگوریتم JSA پایه اعمال گردیده است تا قابلیت‌های جستجوی آن از جمله اکتشاف و بهره‌برداری افزایش یابد. همچنین، در تغییرات اعمال شده تلاش گردیده است تا در کنار افزایش کارایی الگوریتم JSA پایه، قابلیت گریز از بهینه‌های محلی آن نیز تقویت شود. در ادامه این فصل تغییرات اعمال شده بر روی JSA و جزئیات روش پیشنهادی بیان می‌گردند. در گام نخست توسعه روش EMLJA، معادله (۱۳) به صورت زیر تغییر داده شده است:

$$X_i = X_B + \gamma \cdot r_5 \times (ub - lb) \times LV \quad (43)$$

در معادله (۴۳)، LV مقدار به‌دست‌آمده توسط تابع Levy flight است که به صورت ریاضی در معادله (۴۴) فراهم شده است. همچنین در این معادله X_i با X_B جایگزین شده است تا نخبه‌گرایی الگوریتم و توجه آن به بهترین راه‌حل یافته شده تاکنون بیشتر شود.

$$LV = \delta \times \frac{\rho}{|\mu|^{1/\alpha}} \quad (44)$$

$$\rho = Normal(0, \sigma_\rho^2) \quad (45)$$

$$\mu = Normal(0, \sigma_\mu^2) \quad (46)$$

در معادلات (۴۴) تا (۴۶)، δ ضریبی است که مقدار آن برابر σ_μ ، برابر یک، α برابر ۱/۵ بوده و σ_ρ توسط معادله (۴۷) به دست می‌آید.

$$\sigma_\rho = \left(\frac{\Gamma(1 + \alpha) \cdot \sin\left(\frac{\pi \cdot \alpha}{2}\right)}{\Gamma\left(\frac{1 + \alpha}{2}\right) \cdot \alpha \cdot 2^{\frac{\alpha-1}{2}}} \right)^{1/\alpha} \quad (47)$$

در گام بعدی، معادله (۱۴) به صورت زیر اصلاح شده است:

$$X_i = X_B + step \quad (48)$$

معادلات (۴۷) و (۴۸)، قابلیت بهره‌برداری الگوریتم را تقویت کرده و نواحی اطراف بهترین راه‌حل پیداشده تاکنون را به صورت بهینه

جستجو می‌کنند که باعث بهبود فرآیند بهینه‌سازی می‌شود. علاوه بر این، پارامتر کنترل زمان در EMLJA به منظور کاهش بار محاسباتی الگوریتم حذف شده است. در الگوریتم پیشنهادی EMLJA برای تقویت قابلیت اکتشاف نیز از عملگر جهش (mutation) استفاده شده است که به صورت ریاضی در معادله (۴۹) ارائه شده است.

$$X_i^j = X_i^j + \eta \cdot \theta \quad (49)$$

$$j = \{h_i | h_i < \lambda, i = 1, 2, \dots, dim\} \quad (50)$$

در معادلات (۴۹) و (۵۰)، η ضریبی است که مقدار آن برابر ۰/۵ در نظر گرفته شده، θ برداری با اندازه dim است که مقادیر آن به صورت تصادفی از بازه (0,1) انتخاب شده، λ پارامتری است که مقدار آن در روش پیشنهادی برابر ۰/۴ تنظیم شده و h برداری است که مقادیر آن با توزیع نرمال به‌دست آمده است.

الگوریتم (۲): شبه کد مکانیزم جستجوی همسایگی آشوبناک
ورودی‌ها: جمعیت فعلی (X)، بهترین راه‌حل (X_B) و برازندگی آن (X_F)
خروجی‌ها: بهترین راه‌حل (X_B) و برازندگی آن (X_F)
مقدار i را برابر یک کن
تا زمانی که $i \leq K$ است:
CV را توسط معادله (۲) محاسبه کن
X_{r_1} و X_{r_2} را از جمعیت فعلی انتخاب کن
X_B جدید را با استفاده از معادله (۵۱) پیدا کن
برازندگی X_B جدید را محاسبه کن
اگر X_B جدید بهتر از X_B قبلی است، آن را جایگزین کن
بهترین راه‌حل (X_B) و برازندگی آن (X_F) را برگردان

با توجه به این نکته که بهترین راه‌حل یافته شده تاکنون در هدایت سایر راه‌حل‌های جمعیت موجود نقش بسزایی دارد، در الگوریتم EMLJA مکانیزمی به نام جستجوی همسایگی آشوبناک فراهم گردیده است تا پیرامون بهترین راه‌حل به صورت موثری جستجو شود. این مکانیزم که شبه کد آن در الگوریتم (۲) فراهم شده است، از معادله (۵۱) برای جستجوی پیرامون بهترین راه‌حل استفاده می‌کند.

$$X_B = X_B + (CV - 0.5) \times (X_{r_1} - X_{r_2}) \quad (51)$$

بیشتر باشد. سپس، زمان اتمام سرویس‌دهی بر روی منبع جدید نیز محاسبه شده و با مهلت زمانی وظیفه مقایسه می‌گردد. زمان اتمام سرویس‌دهی هر وظیفه باید از مهلت زمانی آن کمتر باشد. در نهایت، بررسی می‌گردد که وظایف پیش‌نیاز برای پردازش این وظیفه قبلاً در این منبع یا منابع دیگر پردازش خواهند شد یا خیر. اگر چنین منبعی برای وظیفه یافته شود، آن وظیفه به صف منبع منتقل می‌گردد. در غیر این صورت، نقشه نگاشت مشمول پناهی می‌شود. در نتیجه، تابع هدف نهایی در الگوریتم EMLJSA به صورت زیر بازنویسی می‌شود:

$$\text{Minimize: } Fit_i = \alpha \cdot T + \beta \cdot E_p + \gamma \cdot E_c + \text{penalty} \quad (55)$$

در نهایت شبه‌کد الگوریتم EMLJSA در الگوریتم (۳) ارائه شده است.

۶- نتایج آزمایش‌ها

برای بررسی اثربخشی نوآوری‌های انجام‌شده و ارزیابی کارایی روش پیشنهادی EMLJSA آزمایش‌ها متعددی انجام‌شده است که این بخش آن‌ها را ارائه می‌کند. چهارچوب استقرار سرویس پیشنهادی در محیط MATLAB نسخه ۲۰۲۱a بر روی سیستمی با پردازنده Intel Core i7، حافظه اصلی ۱۶ گیگابایت و حافظه SSD پیاده‌سازی شده است. سپس، دوازده شبکه با لایه‌های اینترنت اشیا، لایه، مه و ابر طراحی و الگوریتم EMLJSA بر روی آن‌ها اعمال شده است. در این شبکه‌ها که جزئیات آن‌ها در جدول ۱ ارائه شده است، تعداد منابع فراهم شده در هر لایه شبکه و همچنین تعداد و نوع درخواست‌های کاربران متفاوت در نظر گرفته شده تا کارایی الگوریتم به‌طور دقیق‌تری مورد آزمون قرار گیرد.

الگوریتم (۳): شبه‌کد الگوریتم EMLJSA
ورودی‌ها: ub, lb, dim, CV_0, T, N خروجی‌ها: F_B و X_B
برای هر عروس‌دریایی در بازه $i \in [1, N]$ X_i را توسط معادله (۱) مقداردهی اولیه کن برازندگی عروس‌دریایی i ام (F_i) را محاسبه کن بهترین عروس‌دریایی (X_B) را از جمعیت اولیه پیدا کن تکرار فعلی (t) را برابر یک کن

در معادله (۵۱)، CV مقدار آشوبناک است که توسط معادله (۲) به دست می‌آید، X_{T_1} و X_{T_2} نیز دو راه‌حل از جمعیت فعلی می‌باشند که به صورت تصادفی انتخاب‌شده‌اند.

تاکنون هر راه‌حلی که توسط روش پیشنهادی تولید می‌شود پیوسته است درحالی‌که مساله استقرار سرویس‌ها همان‌طور که در شکل (۱) نیز نشان‌داده شده است یک مساله گسسته است. در نتیجه، در الگوریتم EMLJSA معادله (۵۲) به‌عنوان یک فاز میانی بکار گرفته شده تا راه‌حل گسسته معادل راه‌حل پیوسته تولید گردد.

$$X_i = [X_i] \quad (52)$$

در معادله (۵۲)، $[.]$ تابعی است که قسمت اعشار عدد را حذف می‌کند. هم‌چنین، درحین بروز رسانی راه‌حل‌ها یا تولید معادل گسسته آن‌ها، برخی از ابعاد ممکن است از محدوده فضای جستجو فراتر رود. در نتیجه از معادلات (۵۳) و (۵۴) برای بررسی و تصحیح کران راه‌حل‌ها استفاده شده است.

$$X_i^j = \min(X_i^j, ub^j) \quad (53)$$

$$X_i^j = \max(X_i^j, lb^j) \quad (54)$$

با توجه به معادلات (۲۶) تا (۲۹) بیان شد که در مساله استقرار منابع، محدودیت‌های CPU، RAM، دیسک و مهلت زمانی وجود دارد که باید توسط الگوریتم رعایت شود. در الگوریتم EMLJSA اگر راه‌حلی تولید گردد که یکی از چهار محدودیت بیان‌شده در آن رعایت نشده باشد، مکانیزم مهاجرت فراخوانی می‌شود. توسط مکانیزم مهاجرت، سرویسی که یک یا چند محدودیت آن برآورد نشده است، به منبعی منتقل می‌شود که ظرفیت خالی داشته و محدودیت‌های CPU، RAM، دیسک و مهلت زمانی سرویس را برآورد کند. اگر هیچ منبعی وجود نداشته باشد که CPU، RAM و دیسک باقیمانده آن بیشتر از مقدار CPU، RAM و دیسک مورد نیاز سرویس باشد، یا مهلت زمانی آن را برآورد کند، یک عدد بزرگ به‌عنوان جریمه به مقدار تابع هدف آن راه‌حل اضافه می‌شود تا در تکرارهای بعدی راه‌حل دیگری جایگزین آن شود.

در مکانیزم مهاجرت، تک‌تک محدودیت‌های تعریف‌شده در همه وظایف بررسی گردیده و وظایفی که محدودیت‌های آن‌ها ارضا نشده است مشخص می‌شوند. سپس، برای هرکدام از وظایف، مکانیزم به دنبال منبعی می‌گردد که پردازنده، رم و حافظه باقیمانده آن از میزان پردازنده، رم و حافظه موردنیاز برای پردازش وظیفه

- Sine Cosine Algorithm (SCA) [۲۷]
- Tunicate Swarm Algorithm (TSA) [۲۸]
- Whale Optimization Algorithm (WOA) [۲۹]

شایان ذکر است که مجموعه جواب‌های به‌دست‌آمده توسط الگوریتم‌ها توسط نمودارهای جعبه‌ای مقایسه شده‌اند تا محدوده راه‌حل‌های هر الگوریتم به‌طور مناسب مقایسه گردد.

جدول (۱): مشخصات شبکه‌های طراحی و بکار گرفته شده

مقدار	مشخصه
1000 × 1000 m	Deployment area
30 devices	N_I
100-250	N_S
35-60	N_{ES}
15-30	N_{FN}
10	N_{PM}
1.2	T_P
0.4	α
0.3	β
0.3	γ
[300, 500, 800, 1000, 1200] MIPS	R_i^{CPU}
[256, 512, 1024, 1536] MB	R_i^{RAM}
[128, 256, 512, 1024, 1536] GB	R_i^{DISK}
[1000, 2000, 3000, 4000, 5000] MI	S_i
10-30 s	DL_i
[700, 1000, 1500, 2000] MIPS	e_i^{CPU}
[1024, 2048, 3072, 4096] MB	e_i^{RAM}
[512, 1024, 2048, 3072] GB	e_i^{DISK}
2000 MIPS	e_i^{BW}
0.1-0.5 JPMI	e_i^{ECR}
[1500, 2000, 2500, 3000] MIPS	f_i^{CPU}
[2048, 4096, 6144, 8192] MB	f_i^{RAM}
[2048, 3072, 4096, 5120] GB	f_i^{DISK}
1500 MIPS	f_i^{BW}
0.5-1.5 JPMI	f_i^{ECR}
5000 MIPS	c_i^{CPU}
16384 MB	c_i^{RAM}
10240 GB	c_i^{DISK}
1000 MIPS	c_i^{BW}
3.5 JPMI	c_i^{ECR}

به همین ترتیب، مجموعه راه‌حل‌های به‌دست‌آمده توسط الگوریتم‌های به صورت عددی با معیارهای آماری (بهترین، بدترین، میانگین، میانه و انحراف معیار جواب‌های به‌دست‌آمده) با یکدیگر مقایسه شده‌اند. در کنار این، نرخ همگرایی الگوریتم‌ها نیز به‌منظور ارزیابی دقیق‌تر مقایسه شده‌اند. در نهایت، نتایج الگوریتم‌ها با آزمون آماری غیرپارامتری Wilcoxon اعتبار سنجی شده است. همچنین، آزمایش‌ها انجام شده در این پژوهش در قالب سه سناریو سازمان‌دهی شده‌اند. در هر سناریو چهار شبکه با لایه‌های اینترنت

تا زمانی که $t \leq T$ است:

الگوریتم JSA اصلاح شده

برای هر عروس دریایی در بازه $i \in [1, N]$

$rr = rand()$

اگر $rr < 0.33$

XJ_i را توسط معادله (۴) محاسبه کن

در غیر اینصورت اگر $rr < 0.66$

XJ_i را توسط معادله (۴۳) محاسبه کن

در غیر اینصورت:

XJ_i را توسط معادله (۴۸) محاسبه کن

کران‌های XJ_i را بررسی و مقدار هدف را محاسبه کن

عملگر جهش

برای هر عروس دریایی در بازه $i \in [1, N]$

XM_i را توسط معادله (۴۹) محاسبه کن

کران‌های XM_i را بررسی و مقدار هدف را محاسبه کن

مجموعه راه‌حل‌های X ، XJ و XM را ادغام کن

N راه‌حل بهتر را به‌عنوان X انتخاب کن

X_B و F_B را به‌روزرسانی کن

X_B را توسط مکانیزم جستجوی همسایگی آشوبناک بهبود

بده

X_B و F_B را برگردان

از آنجایی که الگوریتم‌های فرا ابتکاری ماهیت تصادفی دارند، در اجرای مستقل خود نتایج متفاوتی برمی‌گردانند. از این‌رو، روش پیشنهادی ۲۰ بار به‌طور مستقل بر روی هر شبکه اعمال شده و مجموعه نتایج آن ذخیره شده است. سپس، مجموعه نتایج به‌دست‌آمده توسط الگوریتم EMLJSA با مجموعه نتایج به‌دست‌آمده توسط الگوریتم‌های رقیب که زیر لیست شده‌اند مقایسه شده است. شایان ذکر است که مقادیر پارامترهای این الگوریتم‌ها مقادیر پیش‌فرضی است که توسط توسعه‌دهندگان آن‌ها در مقاله پایه معرفی شده است.

- Arithmetic Optimization Algorithm (AOA) [۲۴]
- Jellyfish Search Algorithm (JSA) [۲۱]
- Multi-Verse Optimizer (MVO) [۲۵]
- Particle Swarm Optimization (PSO) [۲۶]

جدول ۲ نتایج عددی الگوریتم‌ها را ارائه کرده و آن‌ها را از لحاظ معیارهای آماری (بهترین، بدترین، میانگین، میانه و انحراف معیار) مقایسه می‌کند. همچنین، نمودارهای جعبه‌ای نتایج الگوریتم‌ها در شکل (۲) ترسیم شده است. شکل (۳) نیز نرخ همگرایی الگوریتم‌ها در بهترین اجرای آن‌ها را نشان می‌دهد.

نتایج عددی جدول ۲ بیان می‌کنند که بهترین و بدترین جواب به‌دست‌آمده توسط الگوریتم پیشنهادی در شبکه‌های ۱ تا ۴ از بهترین و بدترین جواب به‌دست‌آمده توسط الگوریتم‌های رقیب در این شبکه‌ها بهتر است. همچنین میانگین و میانه مجموعه جواب‌های به‌دست‌آمده توسط الگوریتم EMLJSA از مقادیر متناظر به‌دست‌آمده توسط الگوریتم‌های مشابه بهتر است. با این حال، در شبکه ۴، بدترین پاسخ به‌دست‌آمده توسط الگوریتم‌های AOA، SCA و TSA با بدترین راه‌حل به‌دست‌آمده توسط الگوریتم EMLJSA برابر است. انحراف معیار نتایج الگوریتم‌ها متفاوت و پراکنده است که در ادامه تشریح می‌گردد.

اشیا، لبه، مه و ابر برای ارزیابی کارایی الگوریتم‌ها استفاده شده‌اند. شبکه‌های هر سناریو از لحاظ یک مشخصه شبکه متغیر و از لحاظ سایر مشخصه‌ها یکسان است. در ادامه، نتایج آزمایش‌ها به تفکیک سناریوها ارائه می‌گردد.

۱-۶ شبکه با تعداد سرویس‌های درخواستی متفاوت

این بخش از آزمایش‌ها به بررسی عملکرد الگوریتم‌ها در شبکه‌هایی با تعداد خدمات مختلف که توسط کاربران شبکه درخواست می‌شود می‌پردازد. در آزمایش‌ها انجام شده در این سناریو، چهار شبکه مختلف با تعداد ۱۰۰، ۱۵۰، ۲۰۰ و ۲۵۰ سرویس درخواست شده در هر تایم فریم طراحی شده است. در این شبکه‌ها تعداد دستگاه‌های اینترنت اشیا، سرورهای لبه، گره‌های مه و ماشین‌های پردازش ابری به ترتیب ۳۰، ۶۰، ۳۰ و ۱۵ دستگاه در نظر گرفته شده‌اند. سایر پارامترهای این شبکه‌ها برابر مقادیر ارائه شده در جدول ۱ است.

جدول (۲): نتایج آماری الگوریتم‌ها در شبکه‌های ۱ تا ۴

شبکه	الگوریتم	معیار آماری			
		انحراف معیار	بدترین	میانه	میانگین
۱	AOA	4.8970E-01	2.4197E+01	2.3632E+01	2.3490E+01
	JSA	4.1235E-01	2.3987E+01	2.3406E+01	2.2272E+01
	MVO	6.1238E-01	2.4474E+01	2.3550E+01	2.2217E+01
	PSO	4.3793E-01	2.4104E+01	2.3424E+01	2.2416E+01
	SCA	5.1897E-01	2.4529E+01	2.3450E+01	2.2580E+01
	TSA	4.4043E-01	2.4114E+01	2.3274E+01	2.2675E+01
	WOA	4.9418E-01	2.4533E+01	2.3457E+01	2.2759E+01
	EMLJSA	5.1150E-01	2.3824E+01	2.2815E+01	2.1929E+01
۲	AOA	1.1032E+00	3.8994E+01	3.7460E+01	3.7375E+01
	JSA	7.1878E-01	3.8383E+01	3.6831E+01	3.6900E+01
	MVO	1.0880E+00	3.9958E+01	3.6990E+01	3.6955E+01
	PSO	1.0439E+00	3.8044E+01	3.6593E+01	3.6527E+01
	SCA	7.6182E-01	3.8377E+01	3.7606E+01	3.7312E+01
	TSA	1.0041E+00	3.9174E+01	3.7149E+01	3.7340E+01
	WOA	8.9151E-01	3.8499E+01	3.7011E+01	3.7193E+01
	EMLJSA	1.0779E+00	3.7984E+01	3.6108E+01	3.6246E+01
۳	AOA	9.8204E-01	4.9917E+01	4.8243E+01	4.8221E+01
	JSA	1.3416E+00	5.0232E+01	4.7741E+01	4.7806E+01

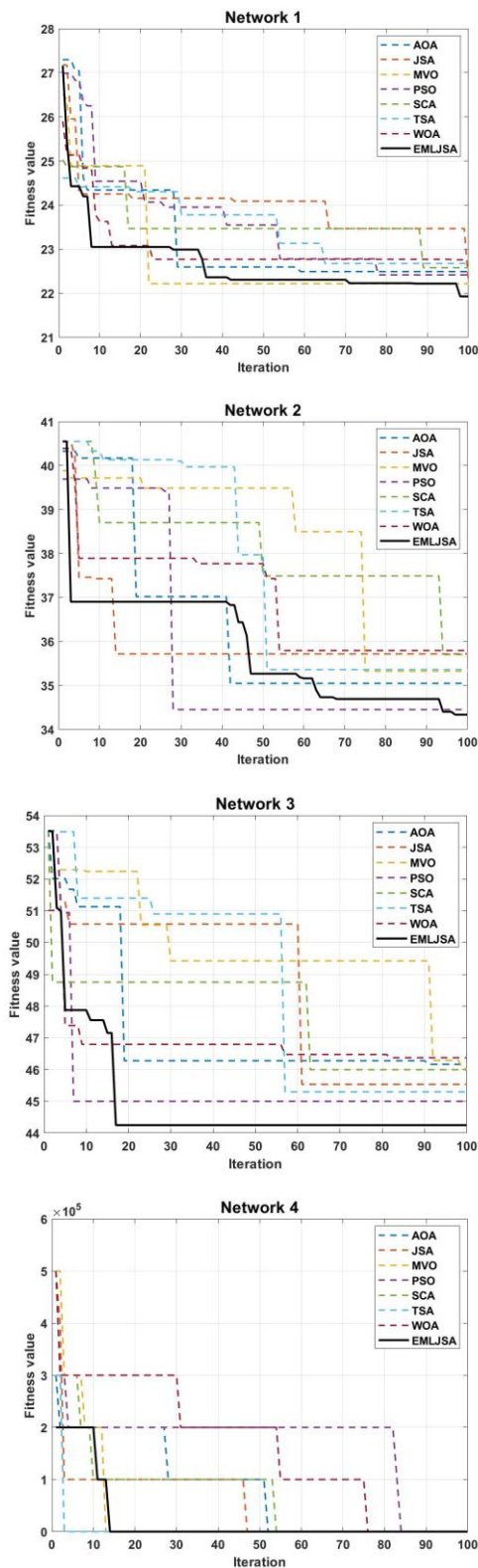
4.6019E+01	4.8247E+01	4.8186E+01	5.0242E+01	1.2232E+00	MVO
4.5002E+01	4.8204E+01	4.8353E+01	4.9908E+01	1.3072E+00	PSO
4.5997E+01	4.8017E+01	4.7905E+01	4.9409E+01	8.8861E-01	SCA
4.5297E+01	4.8192E+01	4.8236E+01	4.9363E+01	9.2490E-01	TSA
4.6372E+01	4.8132E+01	4.8434E+01	5.0139E+01	9.4007E-01	WOA
4.4248E+01	4.6808E+01	4.6903E+01	4.8304E+01	1.0338E+00	EMLJSA
6.9116E+01	8.5067E+04	1.0007E+05	1.0007E+05	3.6633E+04	AOA
6.9051E+01	1.0507E+05	1.0007E+05	2.0007E+05	5.1040E+04	JSA
6.8222E+01	8.0067E+04	1.0007E+05	2.0006E+05	5.2312E+04	MVO
6.6766E+01	8.0066E+04	1.0006E+05	2.0005E+05	5.2311E+04	PSO
7.1027E+01	8.0067E+04	1.0006E+05	1.0007E+05	4.1036E+04	SCA
6.9432E+01	7.5067E+04	1.0006E+05	1.0007E+05	4.4423E+04	TSA
6.8178E+01	7.5068E+04	1.0007E+05	2.0006E+05	5.5009E+04	WOA
5.2485E+01	4.0065E+04	7.3743E+01	1.0007E+05	5.0258E+04	EMLJSA

نمودارهای همگرایی شبکه ۲ مشاهده می‌گردد که همگرایی سریع‌تر این الگوریتم‌ها باعث به دام افتادن آن‌ها در بهینه‌های محلی شده که باعث گردیده الگوریتم دیگر نتواند راه‌حل بهتری را پیدا کند.

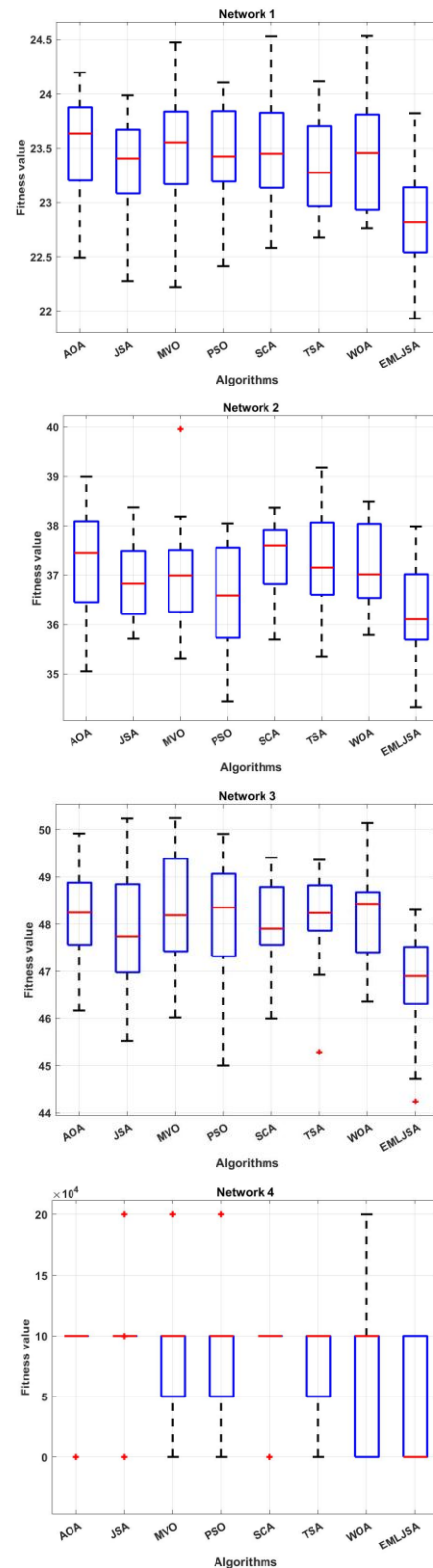
همچنین، الگوریتم EMLJSA نسبت به سایر روش‌ها سریع‌تر به مقدار پایین‌تری از تابع برازندگی همگرا شده است. علاوه بر این، EMLJSA در طول تکرارها مقدار برازندگی پایدارتری داشته و زودتر به مقدار نهایی همگرا شده است. در کنار این، مقدار نهایی تابع برازندگی که توسط EMLJSA به دست آمده است کمتر از سایر الگوریتم‌ها است که نشان‌دهنده کارایی بهتر آن در بهینه‌سازی است. به‌طور کلی، برتری EMLJSA در این نمودار به دلیل همگرایی سریع‌تر و دستیابی به مقدار برازندگی کمتر است.

نمودارهای جعبه‌ای ترسیم‌شده در شکل (۲) نتایج عددی جدول ۳ را تصدیق نموده و نشان می‌دهند که مجموعه جواب‌های به دست آمده توسط EMLJSA از نتایج الگوریتم‌های مشابه بهتر است. همچنین، با بررسی نمودارهای جعبه‌ای شکل (۲) می‌توان دریافت باینکه بازه جواب‌های به دست آمده توسط الگوریتم EMLJSA وسیع‌تر است ولی نوسانات آن در سطوح پایین‌تر تابع هدف است.

نمودارهای همگرایی فراهم شده در شکل (۳) نشان می‌دهند که در شبکه‌های ۱، ۳ و ۴ الگوریتم EMLJSA همگرایی فوق‌العاده‌ای داشته و توانسته است به صورت مناسبی مقدار تابع هدف را کاهش دهد. با این حال، در نمودارهای همگرایی شبکه ۲ مشاهده می‌شود که همگرایی برخی از رقبا از جمله پی‌اس‌او و جی‌اس‌ای از الگوریتم EMLJSA بهتر بوده و در تکرارهای اولیه توانسته‌اند مقدار تابع هدف را با نرخ بیشتری کاهش دهند. با بررسی دقیق‌تر



شکل (3): بهترین نرخ همگرایی الگوریتم‌ها در شبکه‌های ۱ تا ۴



شکل (2): نمودارهای جعبه‌ای نتایج در شبکه‌های ۱ تا ۴

یک روش استقرار سرویس و تخصیص منابع مبتنی بر الگوریتم جستجوی عروس‌های دریایی ارتقا داده شده برای محیط‌های پردازش لبه همراه چهار لایه‌ای

شده‌اند، به ترتیب ۱۵، ۲۰، ۲۵ و ۳۰ گره مه در لایه مه تعبیه‌شده است. همچنین در این شبکه، تعداد دستگاه‌های اینترنت اشیا، تعداد سرویس‌های درخواستی، سرویس‌های لبه و ماشین‌های پردازش ابری به ترتیب برابر ۳۰، ۱۵۰، ۴۰ و ۱۰ دستگاه تعیین شده است. سایر مشخصه‌های شبکه‌های ۵ تا ۸ برابر مقادیری است که در جدول ۱ گزارش شده است. در ادامه، جدول ۳ نتایج الگوریتم‌های استقرار سرویس مورد مطالعه را به صورت آماری مقایسه می‌کند. شکل (۴) نیز مجموعه نتایج هر الگوریتم را توسط نمودارهای جعبه‌ای نشان می‌دهد. همچنین، شکل (۵)، نرخ همگرایی الگوریتم‌ها در شبکه‌های ۵ تا ۸ را نشان می‌دهد.

در نتیجه، با توجه به نتایج آماری جدول ۲، نمودارهای جعبه‌ای شکل (۲) و نمودارهای همگرایی شکل (۳) می‌توان اظهار داشت که الگوریتم EMLJSA در شبکه‌های ۱ تا ۴ از الگوریتم‌های AOA، JSA، PSO، SCA، TSA و WOA عملکرد بهتری داشته است.

۶-۲ شبکه با تعداد منابع مه مختلف

در بخش دوم آزمایش‌ها انجام شده برای ارزیابی عملکرد الگوریتم EMLJSA برای استقرار سرویس‌ها در شبکه‌های چهار لایه‌ای، چهار شبکه دیگر طراحی و شبیه‌سازی شده‌اند که در آن‌ها تعداد گره‌های مه متفاوت است. در این شبکه‌ها که از ۵ تا ۸ شماره‌گذاری

جدول (۳): نتایج آماری الگوریتم‌ها در شبکه‌های ۵ تا ۸

شبکه	الگوریتم	معیار آماری			
		انحراف معیار	بدترین	میان	میانگین
۵	AOA	5.0260E+04	1.0004E+05	4.5872E+01	4.0042E+04
	JSA	5.1297E+04	1.0004E+05	5.0041E+04	4.0004E+01
	MVO	5.1040E+04	1.0004E+05	4.7242E+01	4.5042E+04
	PSO	4.8933E+04	1.0004E+05	4.5805E+01	3.5042E+04
	SCA	5.1039E+04	1.0004E+05	4.6905E+01	4.5042E+04
	TSA	5.1039E+04	1.0004E+05	1.0004E+05	5.5042E+04
	WOA	5.0260E+04	1.0004E+05	4.6338E+01	4.0041E+04
	EMLJSA	2.2360E+04	1.0004E+05	4.1236E+01	5.0412E+03
۶	AOA	4.1421E-01	3.5988E+01	3.5129E+01	3.5228E+01
	JSA	5.7193E-01	3.6133E+01	3.5018E+01	3.5079E+01
	MVO	7.8256E-01	3.7303E+01	3.5627E+01	3.5660E+01
	PSO	5.0398E-01	3.5934E+01	3.5065E+01	3.5036E+01
	SCA	3.9840E-01	3.6097E+01	3.5114E+01	3.5167E+01
	TSA	6.4282E-01	3.6710E+01	3.5398E+01	3.5471E+01
	WOA	7.0752E-01	3.6912E+01	3.5164E+01	3.5317E+01
	EMLJSA	4.1089E-01	3.5912E+01	3.4947E+01	3.4861E+01
۷	AOA	8.3257E-01	3.8716E+01	3.7461E+01	3.7459E+01
	JSA	6.5516E-01	3.7981E+01	3.7223E+01	3.7043E+01
	MVO	7.2144E-01	3.9055E+01	3.7942E+01	3.7861E+01
	PSO	7.2532E-01	3.8522E+01	3.7603E+01	3.7584E+01
	SCA	9.0952E-01	3.9055E+01	3.7434E+01	3.7306E+01
	TSA	5.9252E-01	3.8625E+01	3.7321E+01	3.7285E+01

3.6043E+01	3.7767E+01	3.7942E+01	3.9055E+01	9.3702E-01	WOA
3.4588E+01	3.6728E+01	3.6808E+01	3.8389E+01	1.0164E+00	EMLJSA
3.6636E+01	3.8271E+01	3.8369E+01	3.9410E+01	8.5992E-01	AOA
3.7559E+01	3.8546E+01	3.8470E+01	3.9686E+01	6.1766E-01	JSA
3.6406E+01	3.8429E+01	3.8356E+01	4.0126E+01	8.8857E-01	MVO
3.6200E+01	3.7963E+01	3.7971E+01	3.9670E+01	9.2189E-01	PSO
3.6572E+01	3.8034E+01	3.8241E+01	3.9009E+01	7.1813E-01	SCA
3.6440E+01	3.8127E+01	3.8234E+01	3.9590E+01	7.9704E-01	TSA
3.6524E+01	3.7849E+01	3.7640E+01	3.9310E+01	8.2972E-01	WOA
3.6170E+01	3.7343E+01	3.7306E+01	3.8681E+01	6.2354E-01	EMLJSA

۸

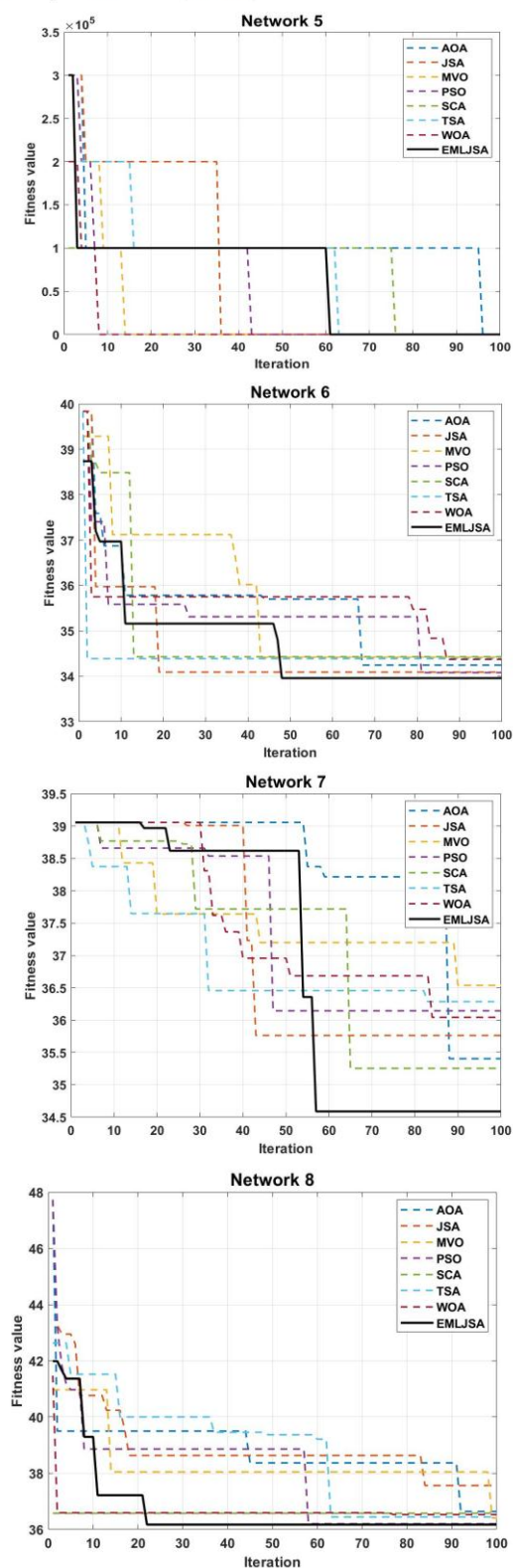
بالاتر آن است. همچنین این نمودارها آشکار می‌سازند که قابلیت‌های اکتشاف و بهره‌برداری روش پیشنهادی در حد مطلوبی بوده و بین آن‌ها تعادل خوبی برقرار است زیرا کاهش مقدار تابع هدف در آن پیوسته و مکرر است. در برخی از شبکه‌ها مانند شبکه ۷ ممکن است چنین برداشت شود که همگرایی الگوریتم‌های رقیب بهتر از روش پیشنهادی است ولی با بررسی نمودارها در تکرارهای نهایی مشخص است که الگوریتم EMLJSA توانسته است از رقبای خود پیشی بگیرد.

علاوه بر این، نمودارهای همگرایی نشان می‌دهند که الگوریتم EMLJSA توان برون‌رفت از بهینه‌های محلی را دارا است. به عنوان نمونه، در تکرار ۳-۴ الگوریتم EMLJSA در شبکه ۵ مشخص است که الگوریتم در بهینه محلی قرار گرفته است. به عنوان مثالی دیگر، در تکرار ۲۳-۲۴ شبکه ۷ عیان است که الگوریتم EMLJSA در دام بهینه محلی افتاده است. با این حال، با بررسی نمودار همگرایی آن می‌توان به سادگی دریافت که در تکرارهای بعدی این الگوریتم توانسته است از بهینه محلی بیرون رفته و مقدار تابع هدف را کاهش دهد.

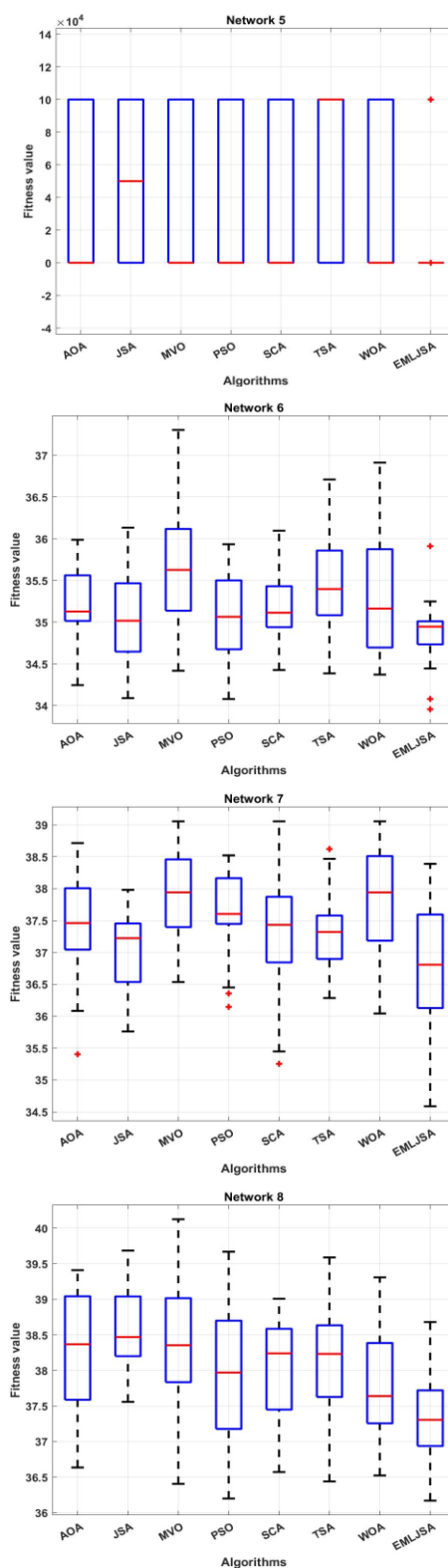
نتایج آماری جدول ۳ آشکار می‌کنند که در شبکه‌های ۵ تا ۸ بهترین راه حل کسب شده برای شبکه‌های ۵ تا ۸ متعلق به روش EMLJSA است. از لحاظ معیارهای میانگین و میانه نیز مشاهده می‌گردد که الگوریتم EMLJSA به صورت چشمگیری از الگوریتم‌های رقیب بهتر عمل کرده و به نتایج بهینه‌تری دست یافته است. از لحاظ بدترین راه حل به دست آمده، تمام الگوریتم‌ها در شبکه ۵ به نتایج یکسانی دست یافته‌اند. در شبکه ۷، بدترین راه حل پیدا شده برای الگوریتم JSA است. در شبکه‌های ۶ و ۸ الگوریتم پیشنهادی از لحاظ بدترین راه حل به دست آمده از رقبای خود پیشی گرفته و بر آن‌ها چیره گشته است.

نمودارهای جعبه‌ای شکل (۴) نتایج آماری جدول ۳ را تأیید کرده و برتری روش پیشنهادی بر الگوریتم‌های AOA، JSA، MVO، PSO، SCA، TSA و WOA را نشان می‌دهند. در این نمودارها مشخص است که مجموعه جواب‌های به دست آمده توسط الگوریتم پیشنهادی در شبکه‌های ۵ تا ۸ در سطوح پایین‌تر تابع هدف قرار دارند که نشان از توانایی بالای روش پیشنهادی در تخصیص منابع و استقرار سرویس‌ها در شبکه‌های ام‌ای سی است.

نمودارهای همگرایی ارائه شده در شکل (۵) نشان می‌دهند که روش پیشنهادی به نسبت به الگوریتم‌های AOA، JSA، MVO، PSO، SCA، TSA و WOA به کمترین مقدار برازندگی در سریع‌ترین زمان رسیده است. کاهش مقدار برازندگی در روش‌های رقیب نیز مشاهده می‌گردد ولی سرعت آن‌ها از روش پیشنهادی کمتر است. این تفاوت سرعت در نرخ همگرایی و پایین‌تر بودن مقدار تابع هدف در روش پیشنهادی نشانه همگرایی بهتر و کارایی



شکل (۵): بهترین نرخ همگرایی الگوریتم‌ها در شبکه‌های ۵ تا ۸

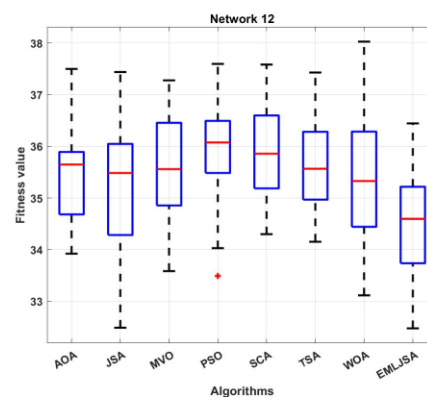
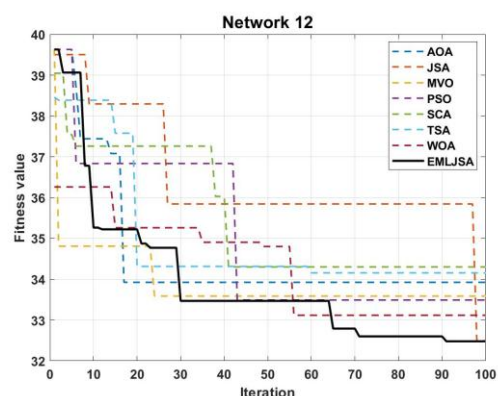
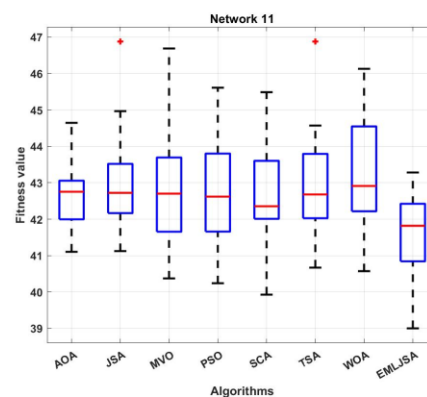
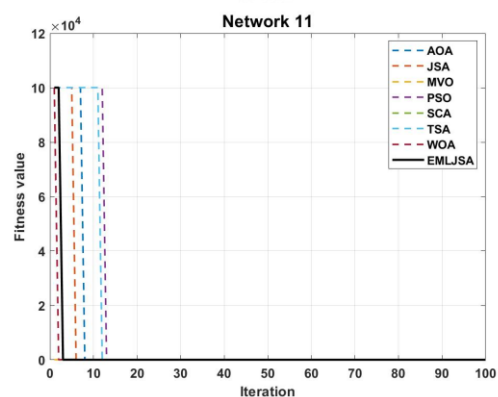
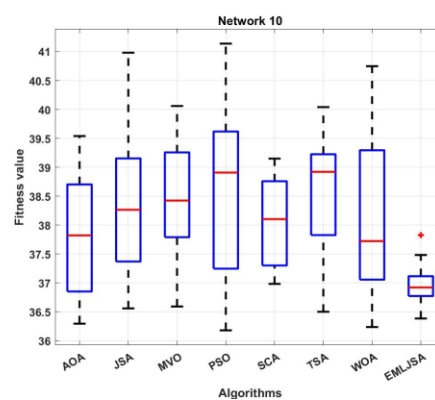
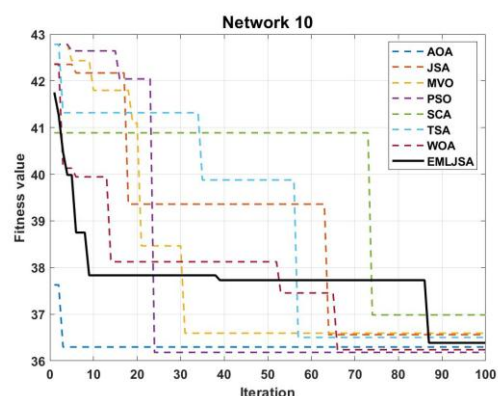
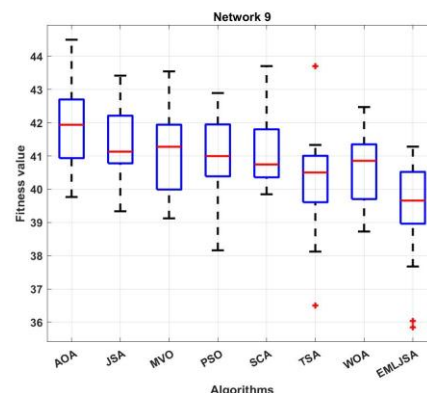
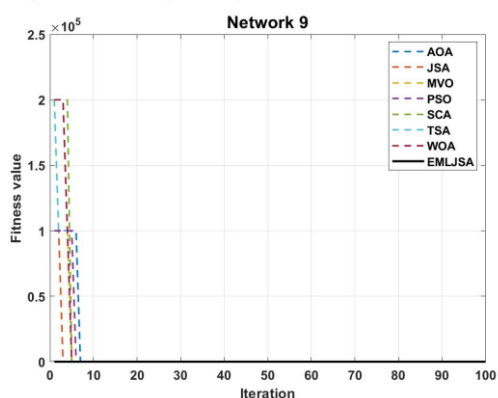


شکل (۶): نمودارهای جعبه‌ای نتایج در شبکه‌های ۵ تا ۸

یک روش استقرار سرویس و تخصیص منابع مبتنی بر الگوریتم جستجوی عروس‌های دریایی ارتقا داده شده برای محیط‌های پردازش لبه همراه چهار لایه‌ای

جدول (۴): نتایج آماری الگوریتم‌ها در شبکه‌های ۹ تا ۱۲

شبکه	الگوریتم	معیار آماری			
		انحراف معیار	بدترین	میانه	میانگین
۹	AOA	1.2535E+00	4.4496E+01	4.1937E+01	4.1784E+01
	JSA	1.1001E+00	4.3416E+01	4.1129E+01	4.1404E+01
	MVO	1.2047E+00	4.3544E+01	4.1276E+01	4.1069E+01
	PSO	1.3286E+00	4.2893E+01	4.0996E+01	4.0901E+01
	SCA	1.0637E+00	4.3702E+01	4.0744E+01	4.1102E+01
	TSA	1.4151E+00	4.3703E+01	4.0502E+01	4.0285E+01
	WOA	1.0303E+00	4.2470E+01	4.0854E+01	4.0568E+01
	EMLJSA	1.5128E+00	4.1280E+01	3.9656E+01	3.9438E+01
۱۰	AOA	1.0392E+00	3.9541E+01	3.7823E+01	3.7856E+01
	JSA	1.2358E+00	4.0982E+01	3.8267E+01	3.8390E+01
	MVO	1.0378E+00	4.0061E+01	3.8426E+01	3.8440E+01
	PSO	1.4687E+00	4.1140E+01	3.8909E+01	3.8532E+01
	SCA	7.4842E-01	3.9151E+01	3.8104E+01	3.8083E+01
	TSA	1.0462E+00	4.0042E+01	3.8922E+01	3.8649E+01
	WOA	1.3508E+00	4.0749E+01	3.7724E+01	3.8185E+01
	EMLJSA	3.4433E-01	3.7831E+01	3.6923E+01	3.6959E+01
۱۱	AOA	9.0599E-01	4.4645E+01	4.2753E+01	4.2669E+01
	JSA	1.3401E+00	4.6876E+01	4.2722E+01	4.2979E+01
	MVO	1.6336E+00	4.6688E+01	4.2701E+01	4.2719E+01
	PSO	1.4604E+00	4.5610E+01	4.2619E+01	4.2779E+01
	SCA	1.2644E+00	4.5487E+01	4.2355E+01	4.2690E+01
	TSA	1.3988E+00	4.6874E+01	4.2678E+01	4.2966E+01
	WOA	1.5351E+00	4.6129E+01	4.2912E+01	4.3266E+01
	EMLJSA	1.1545E+00	4.3282E+01	4.1818E+01	4.1580E+01
۱۲	AOA	9.0163E-01	3.7498E+01	3.5648E+01	3.5503E+01
	JSA	1.4957E+00	3.7438E+01	3.5485E+01	3.5175E+01
	MVO	1.0627E+00	3.7277E+01	3.5559E+01	3.5653E+01
	PSO	9.5977E-01	3.7595E+01	3.6075E+01	3.5912E+01
	SCA	9.6571E-01	3.7583E+01	3.5858E+01	3.5866E+01
	TSA	9.1536E-01	3.7429E+01	3.5567E+01	3.5743E+01
	WOA	1.3701E+00	3.8027E+01	3.5329E+01	3.5393E+01
	EMLJSA	9.8703E-01	3.6444E+01	3.4597E+01	3.4552E+01



شکل (۷): بهترین نرخ همگرایی الگوریتم‌ها در شبکه‌های ۹ تا ۱۲

شکل (۶): نمودارهای جمع‌بندی نتایج در شبکه‌های ۹ تا ۱۲

۳-۶ شبکه با تعداد منابع لبه مختلف

در دو سناریو قبلی، عملکرد روش‌های استقرار سرویس بر روی ۸ شبکه چهار لایه‌ای با تعداد سرویس‌ها و منابع مختلف مورد ارزیابی قرار گرفت که نتایج مقایسه‌ها برتری الگوریتم EMLJSA بر رقبای خود را نشان داد. در این بخش از آزمایش‌ها، چهار شبکه چهار لایه‌ای دیگر برای ارزیابی هرچه بیشتر کارایی روش پیشنهادی طراحی شده است که در آن‌ها تعداد منابع لبه متفاوت است. در شبکه‌های طراحی شده در این سناریو (شبکه‌های ۹ تا ۱۲) به ترتیب ۳۵، ۴۰، ۴۵ و ۵۰ منبع در لایه لبه برای ارائه خدمات به سرویس‌های درخواستی کاربران تعبیه شده است. شایان ذکر است که در شبکه‌های ۹ تا ۱۲، تعداد دستگاه‌های اینترنت اشیا، تعداد سرویس‌های درخواستی، گره‌های میز و ماشین‌های پردازش ابری ثابت و به ترتیب برابر ۳۰، ۱۵۰، ۲۰ و ۱۰ است. سایر پارامترهای این شبکه در جدول ۱ فراهم شده است. به طور مشابه، جدول ۴ نتایج به دست آمده توسط الگوریتم‌ها در شبکه‌های ۹ تا ۱۲ را به صورت آماری ارائه می‌کند. همچنین، نمودارهای جعبه‌ای و نمودارهای نرخ همگرایی الگوریتم‌ها در این شبکه‌ها به ترتیب در شکل‌های (۶) و (۷) نشان داده شده‌اند.

با توجه به نتایج عددی ارائه شده در جدول ۴ می‌توان مشاهده نمود که بهترین راه حل پیدا شده برای شبکه‌های ۹، ۱۱ و ۱۲ برای الگوریتم EMLJSA است. در شبکه ۱۰، بهترین راه حل پیدا شده توسط الگوریتم PSO از بقیه بهتر است. از لحاظ میانه و میانگین نتایج به دست آمده، الگوریتم EMLJSA از همه الگوریتم‌ها بهتر عمل کرده و مقادیر بهینه‌تری در شبکه‌های ۹ تا ۱۲ کسب کرده است. همچنین، بدترین راه حل در بین ۲۰ اجرای مستقل الگوریتم EMLJSA از بدترین جواب پیدا شده توسط الگوریتم‌های رقیب بهتر است.

نمودارهای جعبه‌ای شکل (۶) نتایج آماری جدول ۴ را تصدیق نموده و نشان می‌دهند که الگوریتم EMLJSA به صورت چشمگیری از الگوریتم‌های AOA، JSA، MVO، PSO، SCA، TSA و WOA بهتر عمل کرده و از آن‌ها پیشی گرفته است. براساس این نمودارها می‌توان مشاهده نمود که مجموعه راه حل‌های یافت شده توسط الگوریتم EMLJSA در عین انسجام

دارای مقادیر تابع هدف کمتری در همه شبکه‌های سناریو سوم است.

نمودارهای همگرایی شکل (۷) نشان می‌دهند که EMLJSA با سرعت بیشتری به سمت مقدار بهینه همگرا می‌شود و نسبت به سایر الگوریتم‌ها نوسان کمتری دارد. این موارد نشان‌دهنده قدرت بالاتر روش پیشنهادی، توانایی آن در یافتن پاسخ‌های بهینه و بهینه‌سازی مؤثرتر آن است. علاوه بر این، با توجه به نمودارهای همگرایی شکل (۶) می‌توان استنتاج کرد که روش پیشنهادی توانسته است در مدت زمان کمتری راه حل‌های بهتری با مقادیر تابع هدف کمتر در فضای مساله پیدا کند. این نمودارها همچنین نشان می‌دهند که الگوریتم EMLJSA توانایی گریز و برون رفت از بهینه‌های محلی را دارد.

۴-۶ نتایج آزمون ویلکاکسون

آزمون ویلکاکسون یک آزمون آماری غیرپارامتری است که در پژوهش‌های بسیاری برای ارزیابی مجموعه جواب‌های کسب شده توسط روش‌های مبتنی بر الگوریتم‌های فرا ابتکاری بکار گرفته شده است. این آزمون، دو مجموعه راه حل را به عنوان ورودی دریافت کرده و یک مقدار احتمالی (p-value) برمی‌گرداند که اساس تصمیم‌گیری است. در این آزمون، یک فرضیه به نام فرضیه صفر تعریف می‌شود که بیان می‌کند تفاوت معنی داری بین مجموعه نتایج دو الگوریتم وجود ندارد. سپس مقدار احتمالی گزارش شده توسط این آزمون برای رد کردن یا ناتوانی در رد کردن فرضیه صفر بررسی می‌شود. اگر مقدار احتمالی گزارش شده توسط آزمون ویلکاکسون کمتر از ۰/۰۵ باشد، می‌توان فرضیه صفر را رد کرده و ادعا نمود که بین نتایج الگوریتم‌ها تفاوت معنی دار وجود دارد. ولی اگر مقدار احتمالی گزارش شده بیشتر از ۰/۰۵ باشد، گفته می‌شود که شواهد کافی برای رد فرضیه صفر وجود ندارد.

در این بخش از آزمایش‌ها، نتایج به دست آمده توسط الگوریتم EMLJSA و الگوریتم‌های AOA، JSA، MVO، PSO، SCA، TSA و WOA توسط آزمون ویلکاکسون مورد ارزیابی قرار می‌گیرند. در این راستا، مجموعه جواب‌های الگوریتم EMLJSA در شبکه‌های ۱ تا ۱۲ به همراه نتایج سایر الگوریتم‌ها به آزمون ویلکاکسون داده شده و مقدار احتمالی به دست آمده به تفکیک

سناریوها در جدول‌های ۵، ۶ و ۷ ارائه شده است. در این جدول‌ها،

مقادیر کوچک‌تر از ۰/۰۵ به صورت توپر نوشته شده است.

جدول (۵): نتایج آزمون آماری ویلکاکسون برای شبکه‌های ۱ تا ۴

شبکه				الگوریتم EMLJSA با
۱	۲	۳	۴	
1.7130E-03	2.0633E-02	5.1672E-04	1.8901E-04	AOA
5.1110E-03	1.0046E-01	3.5918E-03	1.1624E-03	JSA
1.6881E-02	6.1953E-02	1.5073E-03	4.0452E-03	MVO
5.1110E-03	4.7813E-01	5.1110E-03	1.9443E-03	PSO
1.0188E-03	3.5918E-03	2.4950E-03	4.0452E-03	SCA
4.5499E-03	5.1110E-03	5.9342E-04	5.1672E-04	TSA
8.0344E-03	1.8675E-02	1.1624E-03	1.1129E-02	WOA

جدول (۶): نتایج آزمون آماری ویلکاکسون برای شبکه‌های ۵ تا ۸

شبکه				الگوریتم EMLJSA با
۵	۶	۷	۸	
8.9180E-04	5.1110E-03	4.3804E-02	4.0452E-03	AOA
1.6286E-04	1.1689E-01	4.1146E-01	1.8901E-04	JSA
1.1624E-03	4.5499E-03	3.1851E-03	7.7959E-04	MVO
2.8209E-03	3.1346E-01	5.1110E-03	4.7858E-02	PSO
1.0188E-03	5.2222E-02	7.9322E-02	5.1110E-03	SCA
1.8901E-04	6.4246E-03	2.2769E-02	9.9964E-03	TSA
1.3245E-03	3.0365E-02	1.0188E-03	1.0046E-01	WOA

جدول (۷): نتایج آزمون آماری ویلکاکسون برای شبکه‌های ۹ تا ۱۲

شبکه				الگوریتم EMLJSA با
۹	۱۰	۱۱	۱۲	
1.4013E-04	6.4246E-03	5.7340E-03	1.5240E-02	AOA
1.2042E-04	3.3845E-04	2.5360E-04	5.2222E-02	JSA
2.8209E-03	2.5360E-04	2.0633E-02	1.8675E-02	MVO
1.2374E-02	8.9180E-04	2.7621E-02	2.4950E-03	PSO
6.8061E-04	2.5360E-04	6.7355E-02	1.3245E-03	SCA
8.5924E-02	1.2042E-04	3.1851E-03	2.2039E-03	TSA
9.9964E-03	3.5918E-03	6.8061E-04	7.3138E-02	WOA

معنادار بین نتایج الگوریتم پیشنهادی و الگوریتم‌های AOA، JSA،

MVO، PSO، SCA، TSA و WOA تشخیص داده است که

با توجه به مقادیر احتمالی گزارش شده در جدول ۵ مشاهده

می‌گردد که در شبکه‌های ۱، ۳ و ۴ آزمون ویلکاکسون تفاوت

نتیجه‌گیری کرد و باید به نتایج آماری، نمودارهای میله‌ای و نمودارهای همگرایی مراجعه نمود.

۷- خلاصه مطالب و پژوهش‌های آتی

در این مقاله الگوریتم فرا ابتکاری جستجوی عروس‌های دریایی بکار گرفته شده و یک چهارچوب استقرار سرویس و تخصیص منابع جدید با نام EMLJSA برای محیط‌های MEC با لایه‌های اینترنت اشیا، لبه، مه و ابر ارائه شده است. هدف اصلی روش پیشنهادی بهینه‌سازی تخصیص منابع با محدودیت‌های مختلف و کاهش بار محاسباتی است. در چهارچوب پیشنهادی، دستگاه‌های موجود در لایه اینترنت اشیا در بازه‌های زمانی مشخص سرویس‌هایی را درخواست می‌کنند. این سرویس‌ها توسط منابع تعبیه‌شده در لایه لبه، مه و ابر فراهم شده‌اند. EMLJSA بهینه‌ترین منابع در دسترس در لایه‌های لبه، مه یا ابر را برای سرویس‌های درخواستی کاربران انتخاب می‌کند. این امر به دلیل وجود محدودیت‌های CPU، RAM، DISK، Deadline و پهنای باند و همچنین پیش‌نیازهای هر وظیفه و مهلت زمانی آن‌ها کاری چالش برانگیز است. پیچیدگی این کار در محیط‌های مقیاس وسیع با تعداد منابع محدود و درخواست‌های زیاد کاربران بیشتر می‌شود. در الگوریتم EMLJSA برخی از پارامترهای الگوریتم جستجوی عروس‌های دریایی پایه باهدف کاهش بار محاسباتی در تکرارهای الگوریتم حذف شده است. سپس، معادلات این الگوریتم در راستای بهبود آن اصلاح شده‌اند. در گام بعدی، یک نسخه از عملکرد جهش که یک عملکرد ژنتیکی است با الگوریتم جستجوی عروس‌های دریایی ارتقا داده شده ترکیب شده است. همچنین، یک مکانیزم جستجوی همسایگی آشوبناک برای بهبود هر چه بیشتر بهترین راه‌حل موجود در جمعیت بکار گرفته شده است. برای ارزیابی کارایی و عملکرد روش پیشنهادی، دوازده شبکه MEC با لایه‌های اینترنت اشیا، لبه، مه و ابر طراحی شده‌اند که مشخصه و پارامترهای آن‌ها متفاوت است. در این شبکه‌ها، تعداد و نوع سرویس‌های درخواستی، تعداد منابع مستقرشده در لایه‌های لبه، مه و ابر و همچنین پهنای باند به صورت متفاوت تعریف شده‌اند. دلیل این امر ارزیابی دقیق‌تر کارایی الگوریتم‌های بوده است. سپس، روش EMLJSA بر روی این شبکه‌ها اعمال

نشانه برتری آن بر الگوریتم‌های رقیب است. در شبکه ۲، این آزمون بین نتایج الگوریتم پیشنهادی و الگوریتم‌های SCA، AOA، TSA و WOA تفاوت چشمگیری را تشخیص داده ولی در تشخیص تفاوت بین الگوریتم پیشنهادی و الگوریتم‌های JSA، MVO و PSO ناتوان بوده است.

نتایج جدول ۶ نیز بیان می‌کنند که در شبکه ۵ آزمون ویلکاکسون تفاوت معنادار بین الگوریتم پیشنهادی و رقبا را تشخیص داده است. در شبکه ۶، این آزمون بین الگوریتم پیشنهادی و الگوریتم‌های AOA، MVO، TSA و WOA تفاوت معنادار را تشخیص داده ولی در تشخیص تفاوت چشمگیر با الگوریتم‌های JSA، PSO و SCA ناتوان بوده است. در شبکه ۷، این آزمون توانسته است برتری الگوریتم پیشنهادی بر الگوریتم‌های AOA، MVO، PSO، TSA و WOA را اثبات کند ولی نتوانسته است بین نتایج الگوریتم پیشنهادی و الگوریتم‌های JSA و SCA تفاوت عمده پیدا کند. در شبکه ۸ به‌جز الگوریتم WOA، آزمون ویلکاکسون توسط مقادیر احتمالی نشان داده است که الگوریتم پیشنهادی از سایر الگوریتم‌ها بهتر عمل کرده و مجموعه نتایج بهتری پیدا کرده است.

نتایج مقادیر احتمالی گزارش‌شده توسط آزمون ویلکاکسون برای شبکه‌های سناریو سوم در جدول ۷ اظهار می‌کنند که در شبکه ۹ به‌غیر از الگوریتم TSA این آزمون بهتر بودن مجموعه پاسخ‌های پیداشده توسط الگوریتم پیشنهادی بر الگوریتم‌های رقیب را تشخیص داده است. در شبکه ۱۰ الگوریتم پیشنهادی از همه رقبا بهتر عمل کرده و به نتایج امیدوارکننده‌ای دست یافته است. در شبکه ۱۱، به‌جز الگوریتم SCA، این آزمون تفاوت معنادار بین نتایج روش پیشنهادی و روش‌های مشابه را تشخیص داده است. در شبکه ۱۲ نیز مقادیر احتمالی نشان می‌دهند که نتایج الگوریتم پیشنهادی به‌طور چشمگیری از نتایج الگوریتم‌های JSA و WOA بهتر است.

در حالت کلی، با توجه به نتایج مقادیر احتمالی گزارش‌شده توسط آزمون ویلکاکسون برای شبکه‌های ۱ تا ۱۲ می‌توان مشاهده نمود که الگوریتم پیشنهادی در ۸۴/۵٪ بر رقبا خود چیره گشته و از آن‌ها پیشی گرفته است. در ۱۵/۵٪ باقیمانده نمی‌توان توسط این آزمون

توزیع نرمال، در نظر گرفتن تفاوت‌های کوچک و سادگی و قابل فهم بودن آن اشاره نمود. نتایج آزمایش‌ها و مقایسه‌ها اثربخشی نوآوری‌های انجام شده و برتری الگوریتم پیشنهادی بر روش‌های مشابه را نشان داد.

در پژوهش‌های آتی می‌توان بجای الگوریتم جستجوی عروس‌های دریایی از الگوریتم‌های فرا ابتکاری دیگری استفاده نمود. ترکیب الگوریتم JSA با الگوریتم‌های دیگر نیز می‌تواند نوآوری خوبی باشد. همچنین، بررسی چگونگی تأثیر تغییرات پارامترهای الگوریتم و انتخاب دقیق‌تر متغیرهای هدف می‌تواند به‌طور مستقیم بر پژوهش‌های آتی تأثیرگذار باشد.

References

- [1] M. Khojand, K. Majidzadeh, M. Masdari, and Y. Farhang, "Controller placement in SDN using game theory and a discrete hybrid metaheuristic algorithm," *The Journal of Supercomputing*, vol. 80, no. 5, pp. 6552-6600, 2024.
- [2] M. H. Shirvani and M. Masdari, "A survey study on trust-based security in Internet of Things: Challenges and issues," *Internet of Things*, vol. 21, p. 100640, 2023.
- [3] M. Abedini Bagha, K. Majidzadeh, M. Masdari, and Y. Farhang, "Improving delay in SDNs by metaheuristic controller placement," *International Journal of Industrial Electronics Control and Optimization*, vol. 5, no. 4, pp. 286-296, 2022.
- [4] M. A. Bagha, K. Majidzadeh, M. Masdari, and Y. Farhang, "ELA-RCP: An energy-efficient and load balanced algorithm for reliable controller placement in software-defined networks," *Journal of Network and Computer Applications*, vol. 225, p. 103855, 2024.
- [5] K. Moghaddasi and M. Masdari, "Blockchain-driven optimization of IoT in mobile edge computing environment with deep reinforcement learning and multi-criteria decision-making techniques," *Cluster Computing*, pp. 1-29, 2023.
- [6] M. Hosseinzadeh et al., "Improved butterfly optimization algorithm for data placement and scheduling in edge computing environments," *Journal of Grid Computing*, vol. 19, pp. 1-27, 2021.
- [7] M. Masdari and H. Khezri, "Efficient offloading schemes using Markovian models: a literature review," *Computing*, vol. 102, no. 7, pp. 1673-1716, 2020.

گرفته و نتایج حاصله با هفت الگوریتم مشابه که مبتنی بر الگوریتم‌های AOA, JSA, MVO, PSO, SCA, TSA و WOA می‌باشند مقایسه شده است. در مقایسه‌ها نتایج الگوریتم‌ها به صورت عددی با معیارهای آماری و به صورت گرافیکی توسط نمودارهای جعبه‌ای و نمودارهای همگرایی مقایسه شده‌اند. همچنین، آزمون غیرپارامتری ویلکاکسون در انجام مقایسه نتایج الگوریتم‌های بکار گرفته شده است. این آزمون برای مقایسه دو مجموعه داده به‌ویژه در شرایطی که فرضیات آزمون‌های پارامتری برقرار نیستند، بسیار مفید است. از جمله دلایل مناسب بودن این آزمون نسبت به آزمون‌های مشابه می‌توان به عدم نیاز به فرض

- [8] S. Li, J. Du, D. Zhai, X. Chu, and F. R. Yu, "Task offloading, load balancing, and resource allocation in MEC networks," *IET Communications*, vol. 14, no. 9, pp. 1451-1458, 2020.
- [9] J. Yang, A. A. Shah, and D. Pezaros, "A Survey of Energy Optimization Approaches for Computational Task Offloading and Resource Allocation in MEC Networks," *Electronics*, vol. 12, no. 17, p. 3548, 2023.
- [10] S. H. S. Ebrahimi, K. Majidzadeh, and F. S. Gharehchopogh, "A principal label space transformation and ridge regression-based hybrid gorilla troops optimization and jellyfish search algorithm for multi-label classification," *Cluster Computing*, pp. 1-45, 2024.
- [11] S. Barshandeh, F. Soleimanian Gharehchopogh, B. Abdolazadeh, S. Gholizadeh, and S. Rasooli Sangani, "A Chaotic-integrated Harris Hawks Optimization Algorithm for Solving Numerical Optimization Problems," *The second conference of electricity, mechanics, aerospace, computer and engineering sciences*, Sydney, Australia, 2023.
- [12] M. Masdari, K. Majidzadeh, E. Doustsadigh, A. Babazadeh, and R. Asemi, "Energy-aware computation offloading in mobile edge computing using quantum-based arithmetic optimization algorithm," 2022.
- [13] R. Ghabousian, Y. Farhang, K. Majidzadeh, and A. Babazadeh Sangar, "Hybrid of particle swarm optimization algorithm and fuzzy system for diabetes diagnosis," *International Journal of Nonlinear Analysis and Applications*, vol. 15, no. 2, pp. 39-46, 2024.

- [14] S. Barshandeh, S. Gholizadeh, S. koulaeizadeh, P. Eskandarian, S. Garah Pasha, and A. Ghaffarpour Rahbar, "MPCASMA: A Multi-Population Chaotic-based Hybrid Algorithm for Global Optimization and Its Application in Feature Selection," The 7th International Conference on Applied Research in Science and Engineering", Aachen, Germany, 2023.
- [15] F. Jafarnejad Rezaiyeh and K. Majidzadeh, "NMFA: Novel Modified FA algorithm Based On Firefly Recent Behaviors," Journal of Advances in Computer Research, vol. 10, no. 4, pp. 51-74, 2019.
- [16] R. Sabzalizadeh, S. Barshandeh, and S. Gholizadeh, "An Invasive Weed Optimization-based Energy and Resource-efficient Workflow Scheduling Algorithm for the Cloud Environment," The 20th International Conference on Information Technology, Computers and Telecommunications", Span, 2023.
- [17] M. Salimian, M. Ghobaei- Arani, and A. Shahidinejad, "Toward an autonomic approach for Internet of Things service placement using gray wolf optimization in the fog computing environment," Software: Practice and Experience, vol. 51, no. 8, pp. 1745-1772, 2021.
- [18] S. Nethaji and M. Chidambaram, "Differential Grey Wolf Load-Balanced Stochastic Bellman Deep Reinforced Resource Allocation in Fog Environment," Applied Computational Intelligence and Soft Computing, vol. 2022, 2022.
- [19] M. Kumar, A. Kishor, J. K. Samariya, and A. Y. Zomaya, "An autonomic workload prediction and resource allocation framework for fog enabled industrial IoT," IEEE Internet of Things Journal, 2023.
- [20] A. Asghari, H. Azgomi, and Z. Darvishmofarahi, "Multi-Objective edge server placement using the whale optimization algorithm and Game theory," Soft Computing, pp. 1-15, 2023.
- [21] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," Applied Mathematics and Computation, vol. 389, p. 125535, 2021.
- [22] H. Hu, Q. Wang, R. Q. Hu, and H. Zhu, "Mobility-aware offloading and resource allocation in a MEC-enabled IoT network with energy harvesting," IEEE Internet of Things Journal, vol. 8, no. 24, pp. 17541-17556, 2021.
- [23] Y.-C. Wu, T. Q. Dinh, Y. Fu, C. Lin, and T. Q. Quek, "A hybrid DQN and optimization approach for strategy and resource allocation in MEC networks," IEEE Transactions on Wireless Communications, vol. 20, no. 7, pp. 4282-4295, 2021.
- [24] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," Computer methods in applied mechanics and engineering, vol. 376, p. 113609, 2021.
- [25] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," Neural Computing and Applications, vol. 27, pp. 495-513, 2016.
- [26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-international conference on neural networks, 1995, vol. 4: IEEE, pp. 1942-1948.
- [27] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," Knowledge-based systems, vol. 96, pp. 120-133, 2016.
- [28] S. Kaur, L. K. Awasthi, A. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," Engineering Applications of Artificial Intelligence, vol. 90, p. 103541, 2020.
- [29] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51-67, 2016.

A Service Placement and Resource Allocation Method based on an Enhanced Jellyfish Search Algorithm for Four-layer Mobile Edge Computing Environ

Mohammadreza Haghighat Afshar¹, Kambiz Majidzadeh^{2*}, Mohammad Masdari³,
Faramarz Fathnezhad⁴

¹ PhD Student, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

² Assistant Professor, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

³ Assistant Professor, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

⁴ Assistant Professor, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

Article Information

Original Research Paper

Received:
2024 November 5

Accepted:
2025 February 15

Keywords:
Mobile Edge Computing, Fog
and Cloud Computing, Service
Placement, Resource
Allocation, Optimization

Corresponding Author*:
kambiz.majidzadeh@iau.ac.ir

Abstract

Mobile edge computing (MEC) is a new networking paradigm that organizes resources in multiple layers to provide high quality services. The resources and services provided in each layer are different regarding the quality, speed, and power. Selecting the most optimal resource among the many resources provided in the edge, fog, and cloud layers for the services requested by the network users under various Quality-of-Service (QoS) restrictions is one of the essential challenges in these networks, which is named Service Placement Problem (SPP). An optimal resource placement algorithm can significantly increase the satisfaction of users and service providers and enhance the overall productivity of the entire system by optimizing communications and reducing service times. Consequently, a new framework called EMLJSA is provided in this paper to solve service placement and resource allocation in four-layer MEC environments. In the EMLJSA, an improved version of Jellyfish Search algorithm (JS) is introduced, in which the basic equations are rectified. Also, a variant of the mutation operator and a chaotic neighborhood search mechanism have been utilized in order to enhance exploration, exploitation, and local departure capabilities. In the next step, twelve four-layer MEC networks are designed to evaluate the effectiveness of the contributions and the proposed algorithm. Finally, the results of the proposed algorithm are compared with seven state-of-the-art methods numerically and visually. The experimental results demonstrate the effectiveness of the innovations and the superiority of the proposed method over the existing methods.

 : 10.22034/ABMIR.2025.22356.1071

E-ISSN: [2821-2037](https://doi.org/10.22034/ABMIR.2025.22356.1071) /© 2025. Published by Yazd University This is an open access article under the CC BY 4.0 License (<https://creativecommons.org/licenses/by/4.0/>).

